仓库（repository）

用来存放项目代码，每个项目对应一个仓库，多个项目对应多个仓库

收藏（star）

常规方便查找

复制克隆项目（fork）

张三/test仓库 李四/test仓库

Forked from 张三/test仓库

Fork

Fork

吧别人的仓库复制一份

仓库独立存在

Watch

类似关注。

事务卡片（issue）

发现代码bug但没有成型代码，需要讨论时用

1.Github访问较慢需要翻墙（shadowsocks）

2.私有仓库只能自己或者只有朋友才有权限操作（私有仓库）

一个git库（对应一个开源项目）

开源项目贡献流程

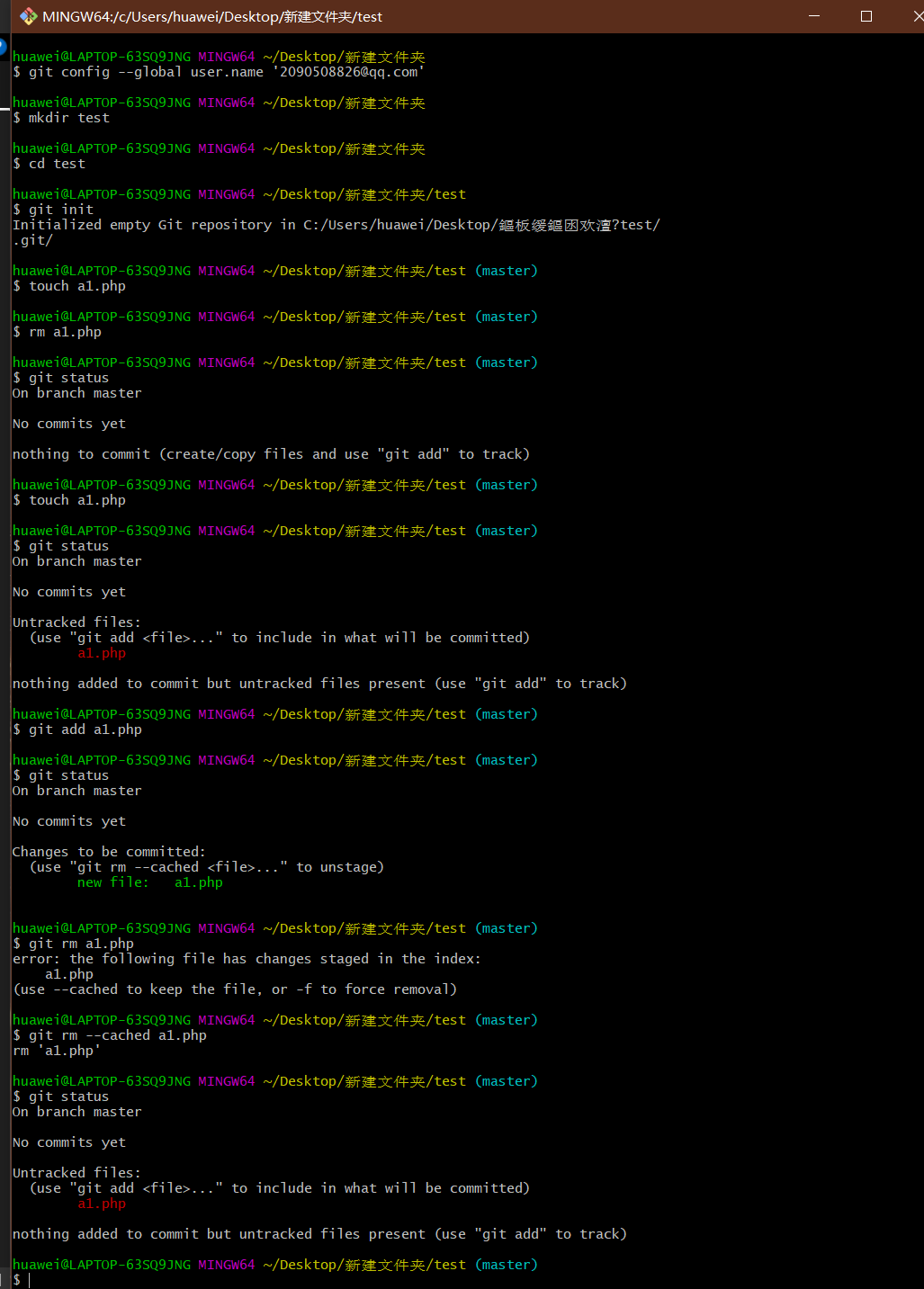
1. 新建issue

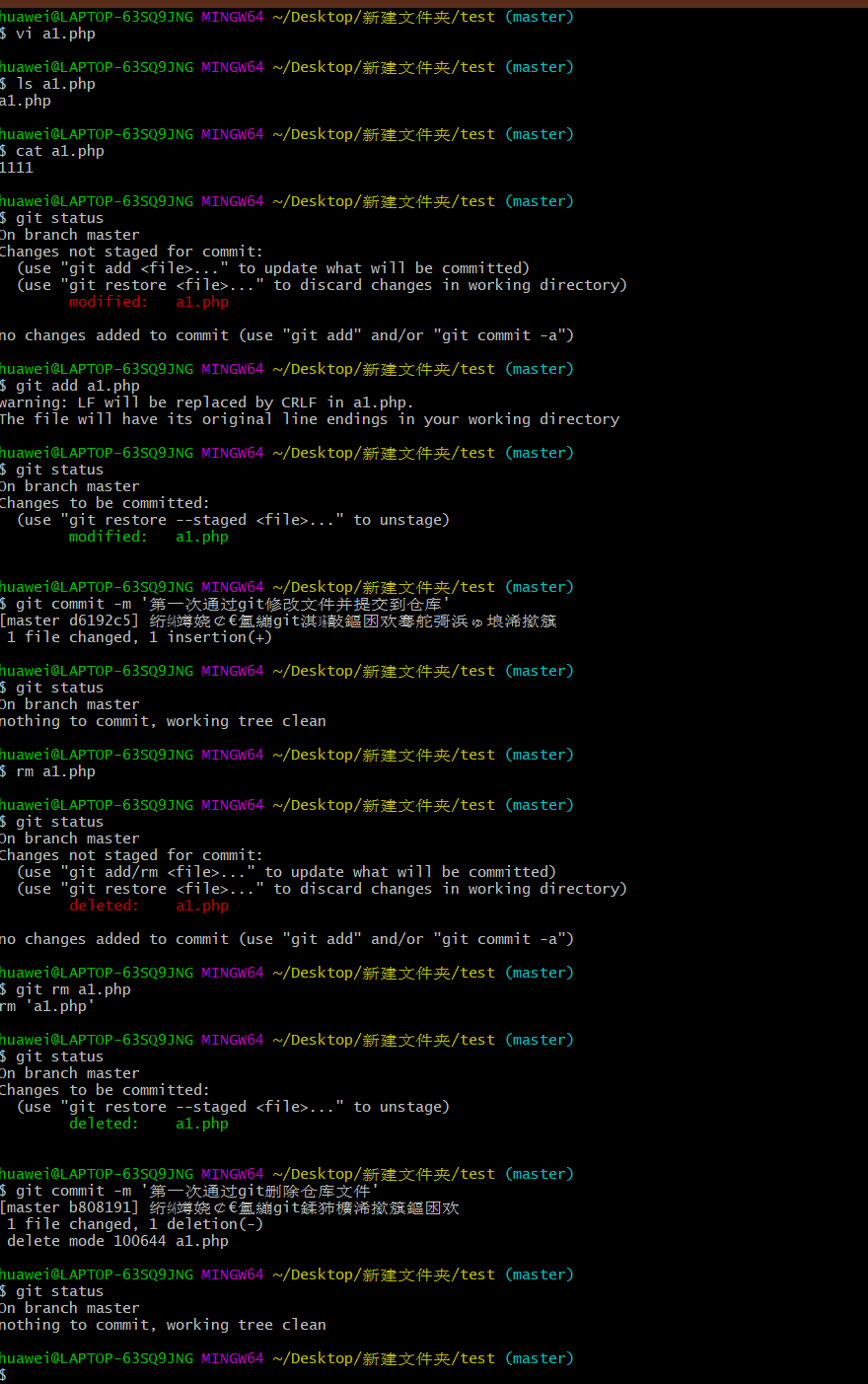
给作者提问题

1. pull request
2. 在自己folk的仓库内修改好，然后pull request

给作者，等待作者合并

Git：





Git提交的是状态

比如删除文件，先把工作区的文件删除然后删除这个change就会自动到暂存区，然后再把暂存区的这个状态通过命令提交到仓库。就完成了一个仓库文件的删除

有两种写法

第一种（我下面先写的）

1.rm a1.php

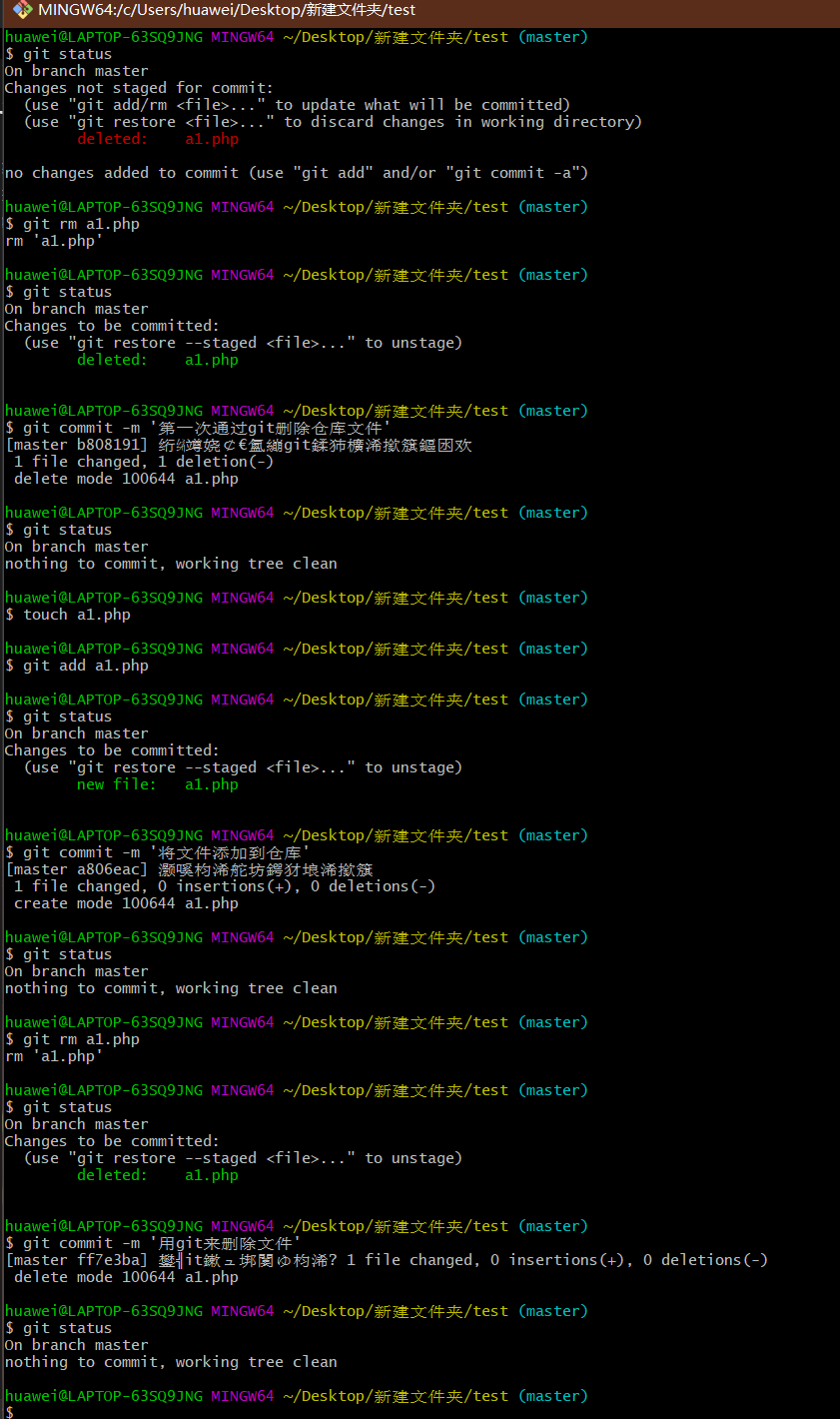
2.git rm a1.php

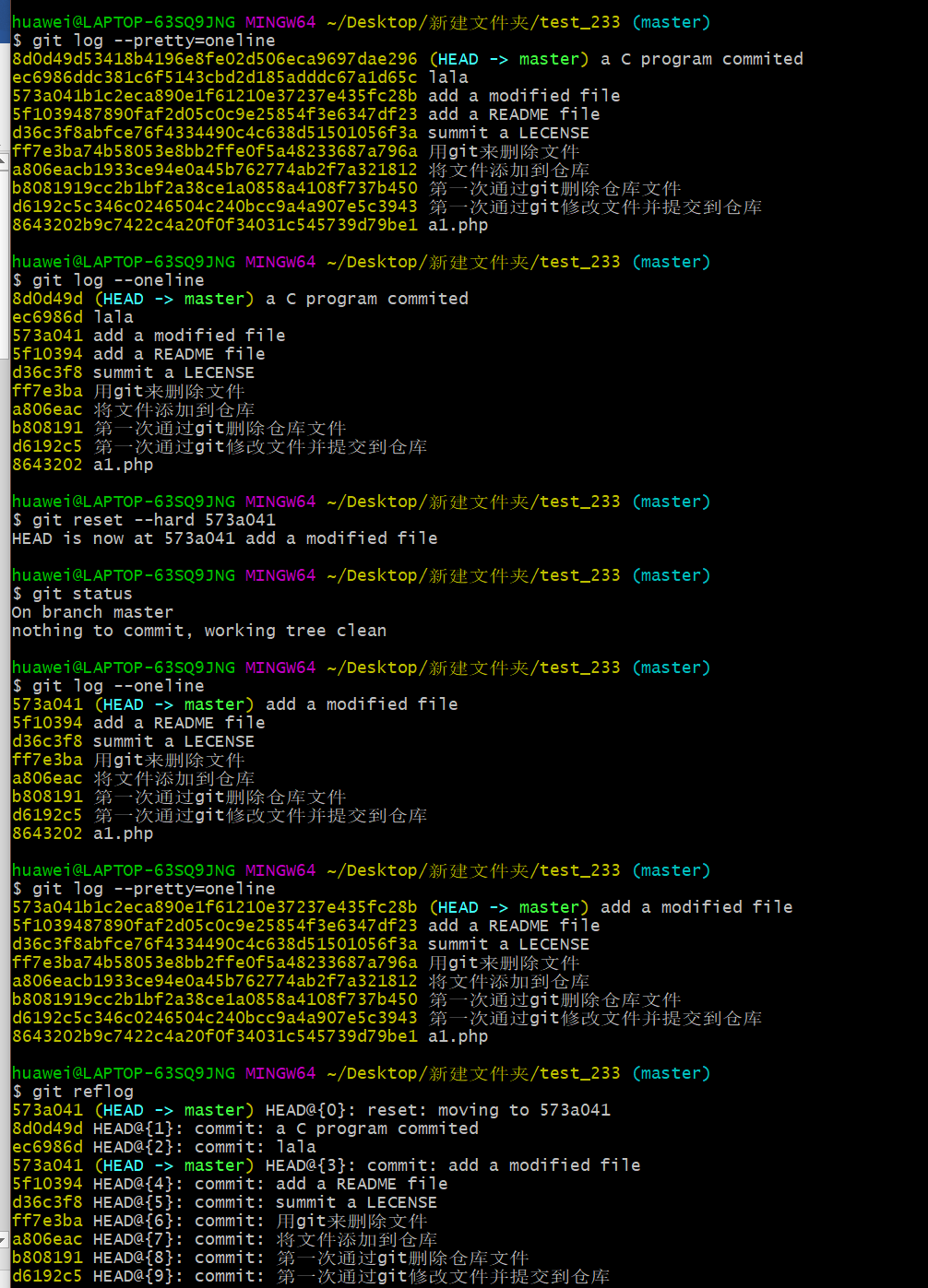
3.git commit -m ‘注释’

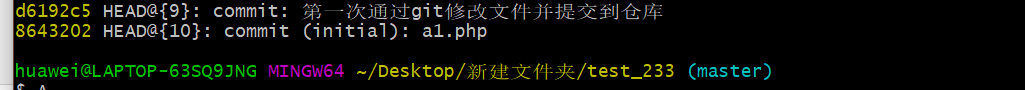
第二种（后写的）

1.git rm a1.php

2.git commit -m ‘注释’







有三种方法调用记录

1. git log ---- 最完整冗长的历史
2. git reflog ----有未来也有过去
3. git log --pretty=oneline ----只有过去

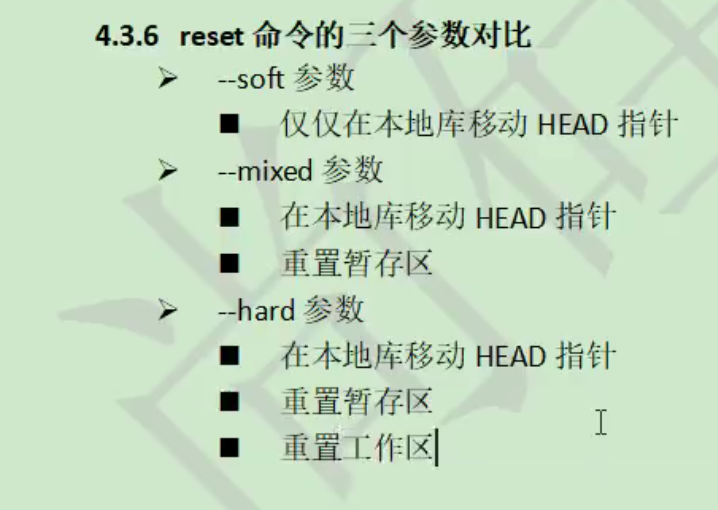
**只有一行显示**

1. git log --oneline ----只有过去（简化版：简化了哈希值）

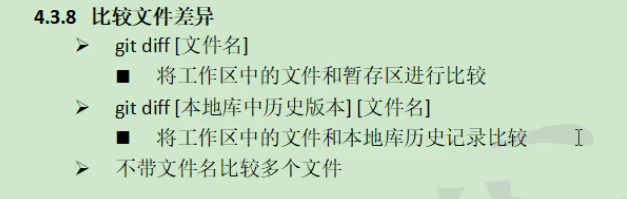
git help [具体命令] --查看该命令的详情

index file 暂存区

working tree 工作区



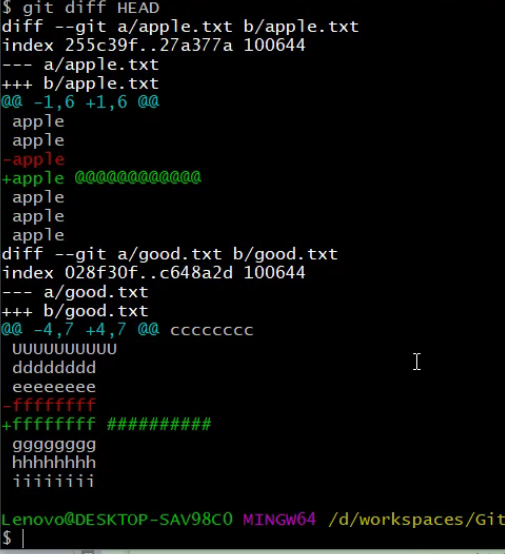
Git diff :



如果加HEAD但不指定文件名即只有

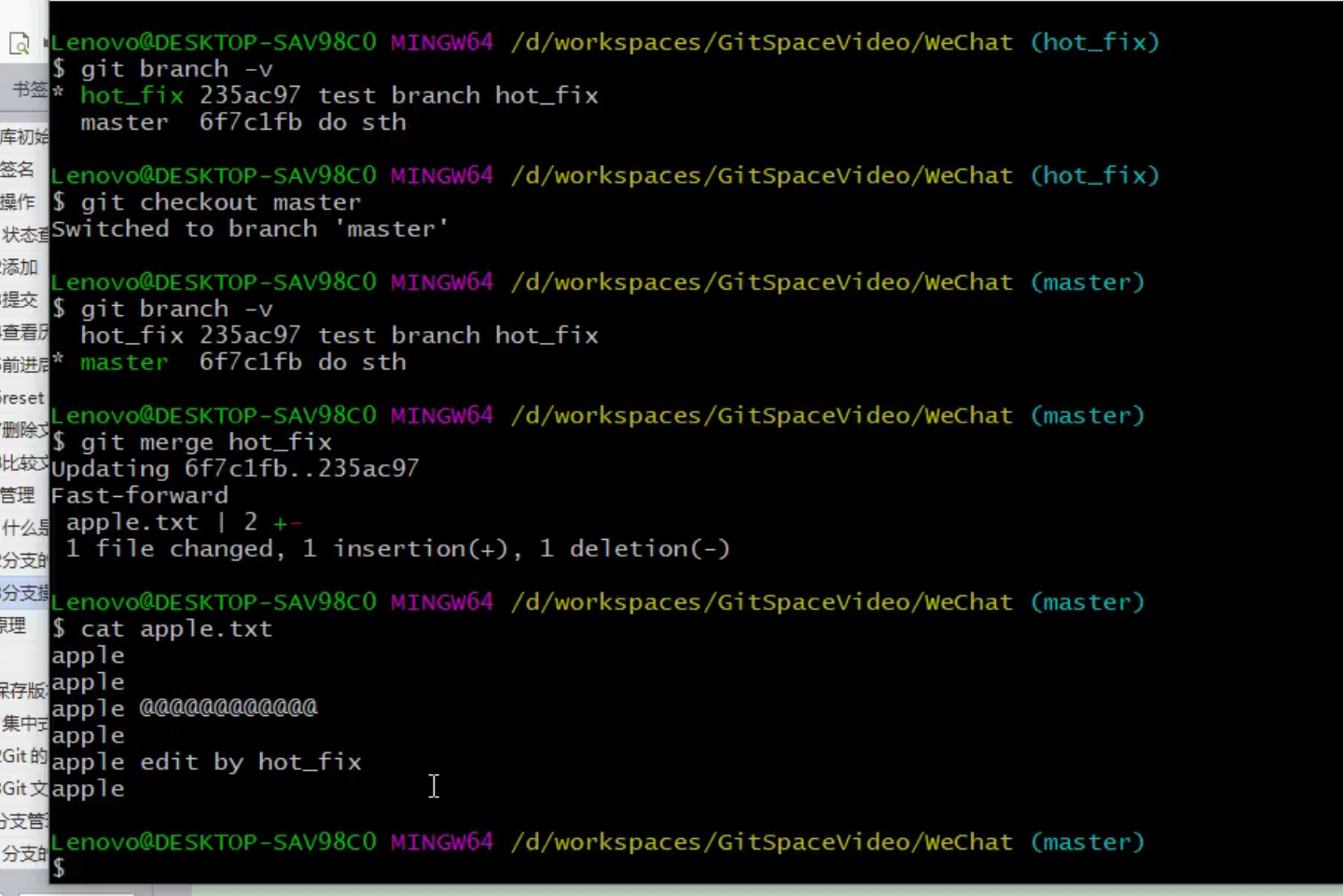
Git diff HEAD

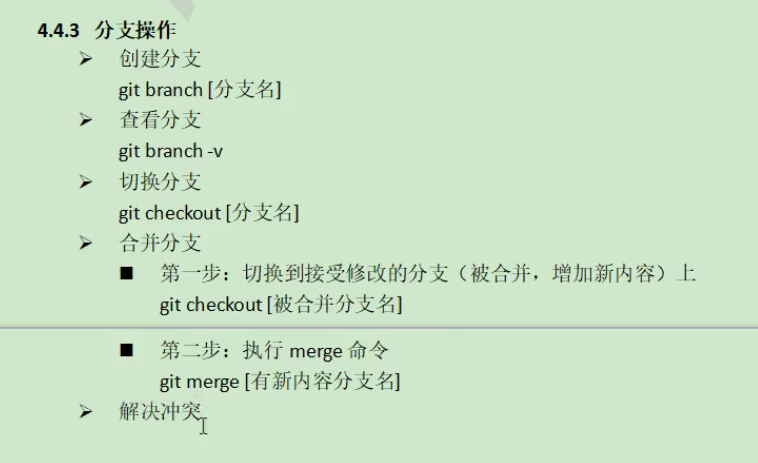
则会把工作区的所有文件和暂存区的所有文件进行比较



但是如果只有git diff 则只会比较最近的修改的文件

分支的命令





分支冲突（两个分支 master 和 hot\_fix 同时发生了修改，并且修改内容产生了冲突）

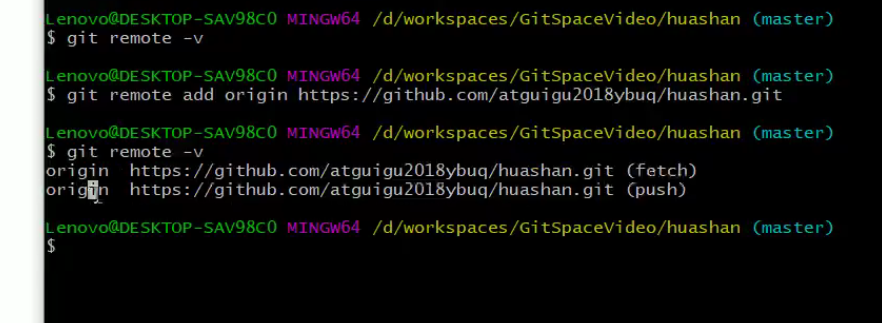


Git 中会选择把所有的问题标记出来然后让用户手动修改

为地址起的别名

为地址起的别名

远程仓库推送的地址保存



从github上复制的地址。

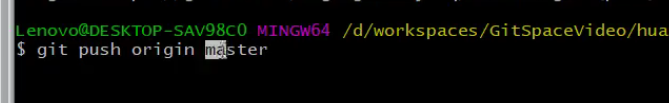
Git remote -v 查看是否有保存的地址

Git remote add origin [地址] 添加合法地址

分支名

别名

推送：



克隆



然后修改完了再推送

Ps. 这里要注意一定要让Gravity邀请Ironman进入小队这样才能允许Ironman把修改晚的东西提交到Gravity的库中

拉取（pull=fetch+merge）

Git fetch [远程地址别名] [远程地址分支名]

Git merge [远程地址别名/远程地址分支名]

好处：

可以fetch之后，通过

Git checkout [远程地址别名/远程地址分支名] 命令

将分支转为拉取下来的分支；

然后用

Cat + [文件名]

查看合并后的文件

如果只用pull那么就直接合并了

Git pull [远程库地址别名] [远程地址分支名]

多人合作的矛盾

发生背景

a拉取了远程库的文件

b提交了自己修改完的文件

a无法再提交自己修改的文件（因为自己修改的文件和远程库中已有的文件有部分行存在矛盾）

a需要先将远程库中的文件拉取下来然后进行矛盾的手动合并，再进行提交推送

结束；；；

还差gitlab服务器没有学习。。

改天学，这个先发上去吧。