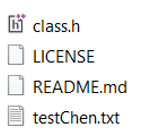
工作区

working dictionary



本地版本仓库 repository



暂存区

stage

本地分支

例: master

橙色框里都在本地；

红色字体为语句

add

commit

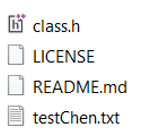
checkout--

reset HEAD

reset --hard HEAD

远端仓库

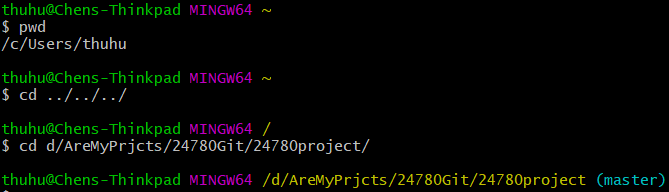
rm



换分支checkout

**前提：**

下文所有操作前先cd 到工作区所在地址：

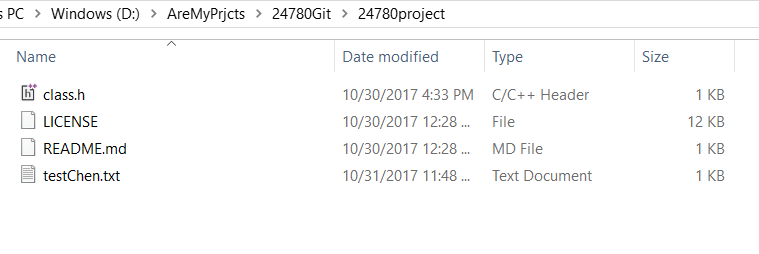


我的工作区地址：D:\AreMyPrjcts\24780Git\24780project

**工作区（working directory）：电脑中你自己指定的目录（下图）**

直接点开文件修改后保存——对应——工作区的修改

-- 将工作区的修改全部取消（成为和暂存区一样的状态P.S.如果没有暂存过，就和本地版本库一样；没有本地版本库，我还不知道，也许和上次从远端仓库clone下来后的状态一样？）



**版本库的暂存区（stage）：用来暂时存放一些本地修改**

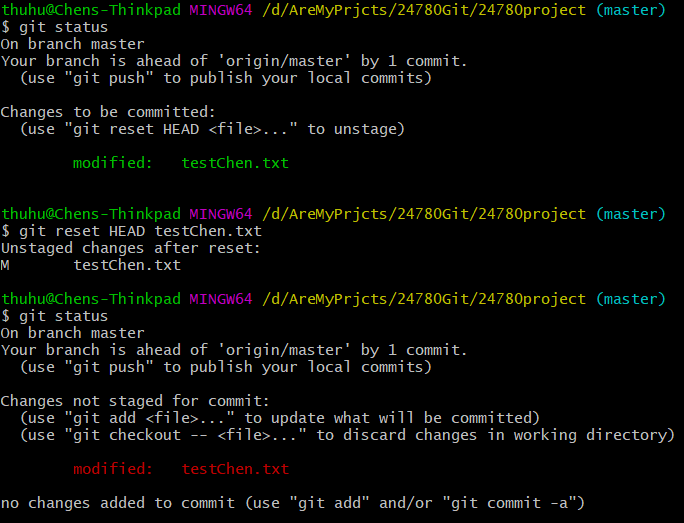
add 语句, rm语句, reset HEAD 语句——对应——暂存区的修改

将工作区某文件添加（或修改）到暂存区

将暂存区某文件从暂存区删除，不影响工作区

查看工作区与暂存区之间testChen.txt文件的差别

将暂存区清空，但不修改工作区内容，见下图（status用法见备注）



此处，原本工作区与暂存区的内容一致（未提交到本地分支），执行reset HEAD filename后，暂存区清空，但工作区不变，因此工作区与暂存区不一致了。如果工作区也想清空，用上一段说到的 git checkout --filename。

**版本库的本地分支（例如：master）：本地版本库的最终版**

commit语句， reset --hard HEAD 语句——对应——本地 “现在” 的分支的修改（初始时默认本地只有一个master分支，但可以自由加减分支）

- 将暂存区的内容同步（修改、添加、删除等各种）到本地分支，对本次同步加上评论，并删除暂存区的内容

-- 退回往上数第一个版本，往上数两个，往上数73个

退回版本后也可以回到修改后的版本，但需要找到版本号，没仔细看怎么搞

**关于分支（branch）：用于本地版本库中的分叉与合并**

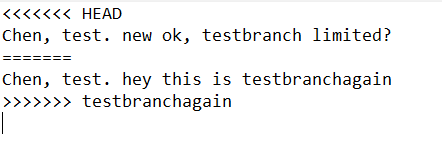
查看本地分支，名字前带\*的是现在工作的分支（HEAD指向的分支）

以目前分支为基础，新建本地分支，名为testbranch，若重名则不执行

切换到testbranch分支

- 新建本地分支，名为testbranchagain，并切换到该分支

将testbranch的内容合并到现在指向的分支。如果没有充足，可以自动合并，如果版本出现冲突，会将两个版本的内容全部放到本地工作区的该文件中（下图），由我们手动选取，然后再执行add, commit操作完成合并步骤。在commit之前，不允许切换分支。



- 删除分支testbranch。不可以在这个分支删除这个分支本身。如果被删除分支已经commit 然而没有merge 到别的分支，而确认真的要删除，那么使用大写D强制删除，即： -

- -查看之前各种分支merge的线路，

之前的commit语句是将暂存区的内容同步到 “现在” 的分支（HEAD指向的分支），reset --hard HEAD语句也是在 “现在” 的分支退回前一个版本。对工作区、暂存区的所有操作都一样使用，切换分区会导致工作区与暂存区的内容丢失，因此在commit之前使用checkout会被拒绝执行。

一般使用自建的本地分支工作，工作到可以使用后，合并到本地的master分支。这样可以保证本地的master分支始终是可以使用的版本。

——————————

—— 分割线，以上操作一切修改都在本地，下面是对远程端的操作，请谨慎

—— 本地操作无需联网

—— 在与GitHub上的远程库沟通之前，需要添加 SSH key

——————————

**远程仓库（默认名叫origin）：团队成员共享的仓库**

clone语句, git push语句

在本地建立一个以项目名为名的文件夹，并将远程仓库内容全部复制下来。如果本地有一个非空的同名文件夹，操作将不会执行。也可以使用

将本地的master分支推送到远程仓库（origin）的master分支中去，也可以推送其他的分支到远程仓库的同名分支中

将远程库的 ChenHu分支同步到本地库，本地会新建一个名叫Chen的分支。如果本地已经有一个想要合并的版本库，可以使用 ----=，将本地叫做ChenHu的分支与远程库上ChenHu分支关联，再进入本地ChenHu分支执行

**备注：**

查看所有区域之间的差别：

{

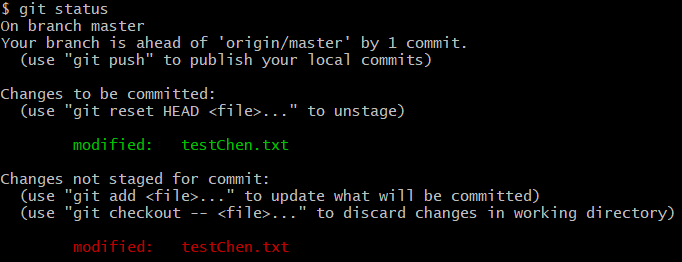
此前我曾顺序执行：

修改testChen.txt，$ add，$ commit； //第一次修改被加入本地分支

再次修改testChen.txt，$ add； //第二次修改在暂存区中

再再次修改testChen.txt； //第三次修改在工作区

}



本地分支与

远程版本库

的差别

在暂存区，未加入本地分支

工作区已修改，未加入暂存区