开发中会一定用到的git所有知识，深入理解，吐血整理

如果连本地操作和分支概念都不熟，就先在这几篇入个门~

【必看】https://mp.weixin.qq.com/s/Hyb2qoL7uKVUVXeme05K3g

【必看】https://mp.weixin.qq.com/s?__biz=MzU0OTE4MzYzMw==&mid=2247491939&idx=3&sn=76ad3a55a96c6f28e8ee354cfe938ed3&chksm=fbb1689dccc6e18b0542449d56b64718e351208bb4b63f897691b7f482634d7375b3f74807f7&scene=27

【必看】https://www.javaclub.cn/tool/60847.html

https://blog.csdn.net/m0_48651355/article/details/120646834

分支管理、合并冲突与解决、拉取推送到远程仓库的冲突与解决

git-merge时到底发生了什么?

一般来说，如果有两个本地分支例如master和dev，未完成开发时始终保持dev提交的指针在master前，master不要有新的提交。dev上完成开发后，切换到master分支【git checkout master】，再执行合并分支的命令【git merge dev】，这样master和dev就合并成功，新的提交节点版本中，有master基础上dev更新的内容（这种情况属于Fast-forward，一定不会产生合并冲突），master后移到和dev指向同一个commit节点。然后本地的dev分支就可以删了。

参考博客:

【必看】详解git merge命令应用的三种情景*相关技巧*脚本之家 (jb51.net)

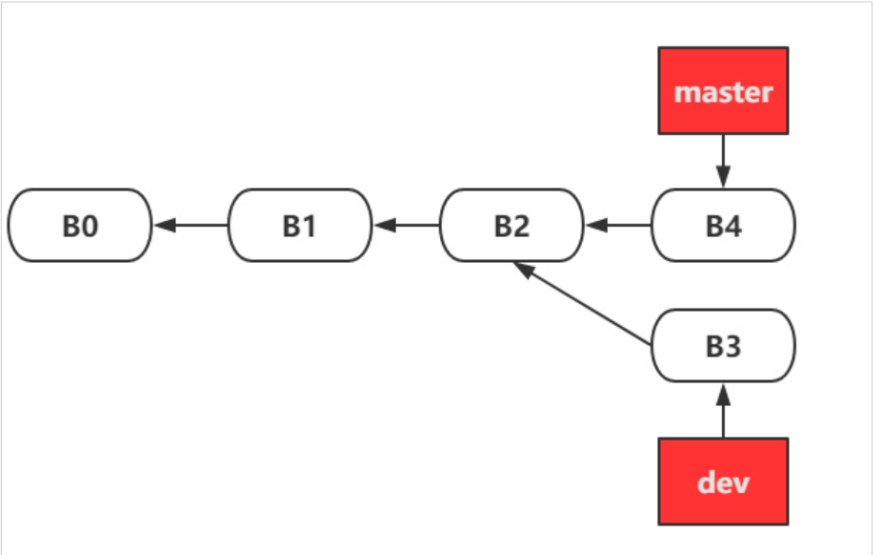
git merge 的三种情况 - 知乎 (zhihu.com)

merge合并产生的冲突什么时候会发生?

只要两个分支都分别提交了若干次（也就是不属于上述Fast-forward的情况），且merge时同一个文件同时被两个分支的commit修改，就会产生冲突（除非在两个分支修改后同一个文件内容还是一模一样的）；如果修改的是不同文件，就不会产生冲突。

1.2 非“快进”，修改不同文件。(无冲突)

当在新分支 dev 进行了一次提交B3，再回到分支 master 又进行一次提交 B4。

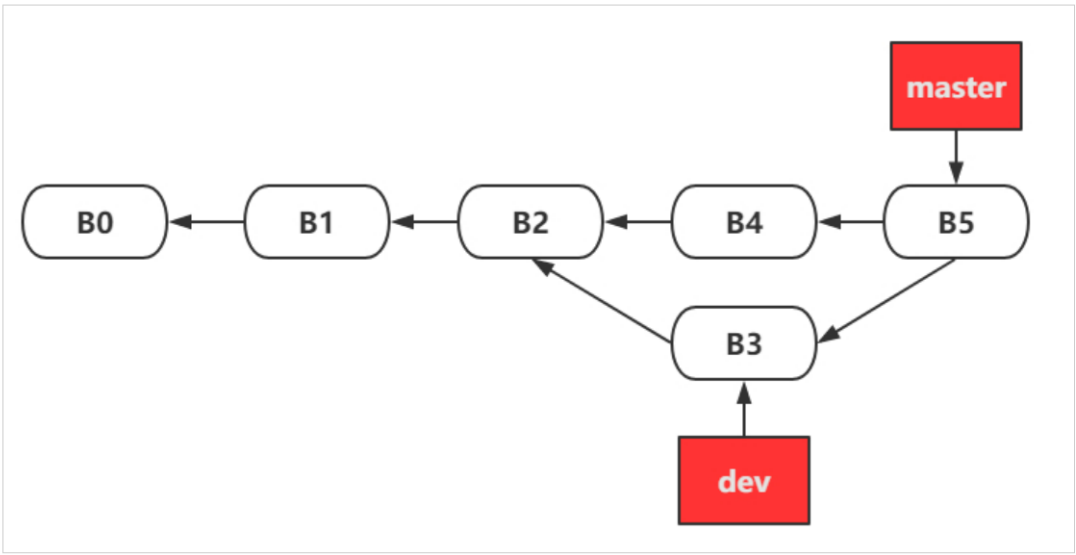


这里用 git merge 合并分为两种情况，现在讲第一种情况：

在 master 分支和 dev 分支的公共祖先 B2 后，master 和 dev 的提交是对不同文件或者同一文件的不同部分进行了修改，Git 可以合并它们。（比如说原来有 test-1 和 test-2 两个文件，B4修改的是 test-1 文件，而B3修改的是 test-2 文件，然后合并两个分支。）

合并是成功的。

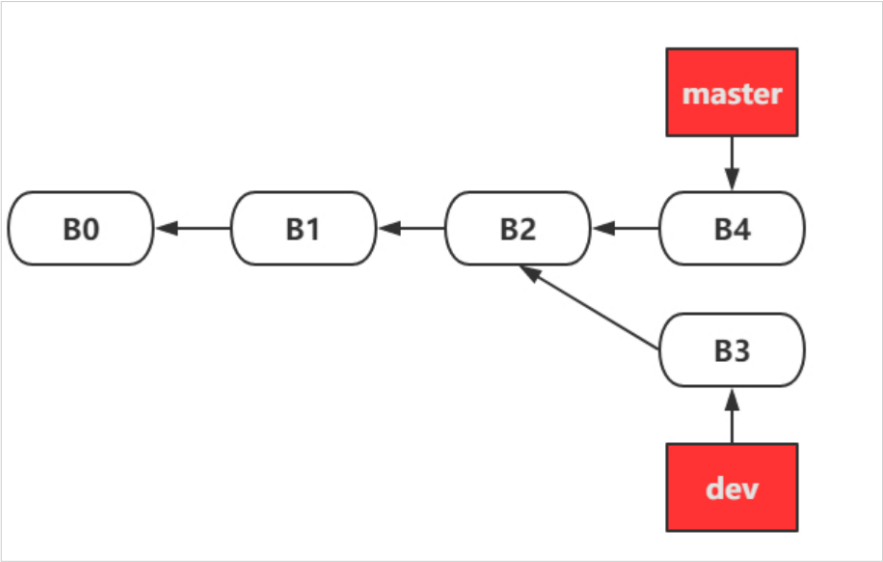
出现这种情况的时候，Git 会使用两个分支的末端所指的快照（B3 和 B4）以及这两个分支的公共祖先（B2），做一个简单的三方合并。注意这里合并后 master 自动 commit 提交了一次，产生了提交B5。而B5中的结果是三方合并的结果。合并结果如下：



最后，合并完成，你已经不再需要dev分支了。 现在你可以删除这个分支。

1.3 非"快进"，修改相同文件。(有冲突)

当在新分支 dev 进行了一次提交B3，再回到分支 master 又进行一次提交 B4。



上面讲的是第一种情况，现在讲第二种情况：

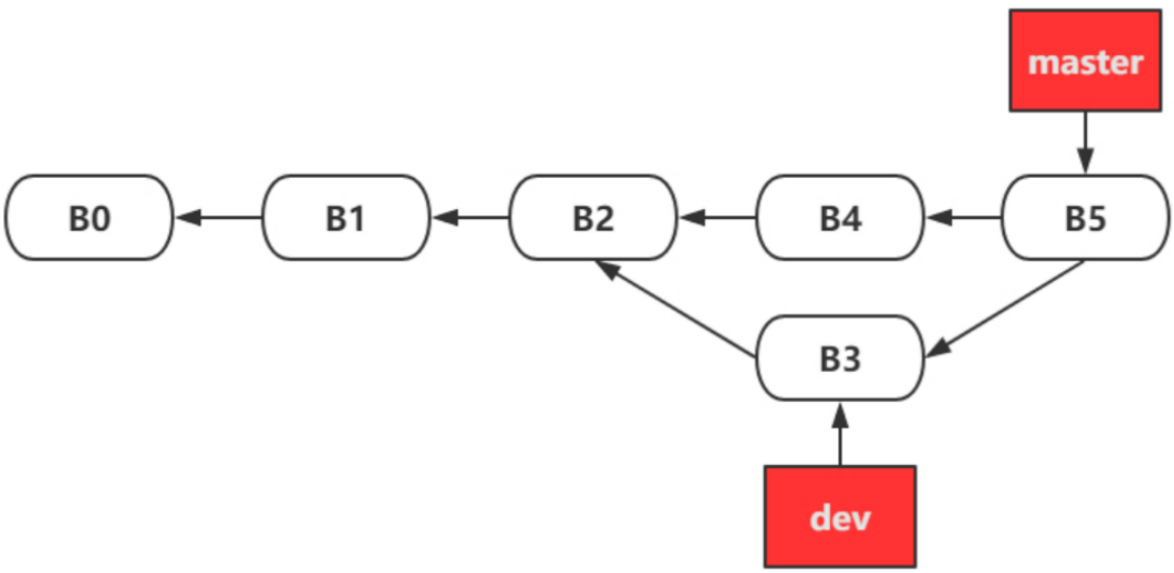
在 master 分支和 dev 分支的公共祖先 B2 后，master 和 dev 的提交是对同一个文件的同一个部分进行了不同的修改，Git 就没法干净的合并它们。（比如说原来有 test-1 和 test-2 两个文件，B4修改的是 test-1 文件，而B3修改的也是 test-1 文件的同一部分，然后合并两个分支。）

merge合并产生的冲突怎样才算解决?

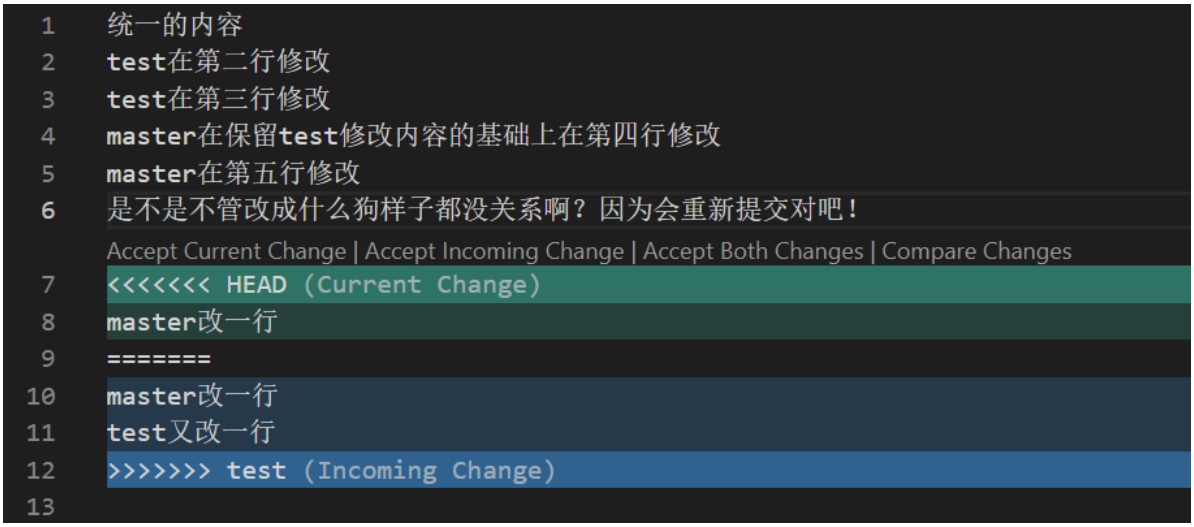
git merge如果发生上述情况的冲突后，git会自动把冲突文件的冲突之处变成这种形式：

```
This is test-1.
update test-1.
add test-1.
<<<<<<< HEAD
test master.
=======
test dev.
>>>>>>> dev
```

这时候只要修改了文件（甚至你不修改，直接留下这些奇怪的符号也行）并git commit一次，冲突就算解决了并合并成功了，此时合并后的提交树是这样的：



至于怎么修改，保留哪部分，完全看你自己，git是不会管的。vscode中给了几种合并冲突的策略供参考，但其实也可以不管他，自己直接随便改。



参考博客：

详解git merge命令应用的三种情景*相关技巧*脚本之家 (jb51.net)

(19 封私信 / 8 条消息) Git到底什么情况下会产生合并冲突？ - 知乎 (zhihu.com)

本地开发完一版新代码需要推送到远程仓库，但远程仓库也被别人更新了一版新代码，此时需要先拉再推【git pull (或git fetch -> git merge origin/master) ->解决冲突->git push】如何理解整个流程?

由于git pull就是git fetch + git merge origin/master，这里只分析git fetch、git merge origin/master和git push

以主分支master为例，首先需要理解"远程仓库的分支master、本地的远程分支origin/master、本地的分支master"之间的关系，才能明白git fetch到底干了什么

先看这篇：

【必看】(92条消息) git 远程分支与本地分支*本地分支和远程分支*滨边美波她男友的博客-CSDN博客

重点总结：

git branch -vv 用于查看本地分支(e.g. master)和本地的远程分支(e.g. origin/master)的关系，注意origin/master的内容不会随着真正远程仓库master的内容改变，需要git fetch才能把这两个内容同步

git branch -a用于查看有哪些本地分支和本地远程分支

git remote show origin命令通过连接网络，去获取本地分支与真正的远程分支的关联情况

再强调两个地方：

1. 本地的远程分支origin/master和远程仓库中的master分支是两个不同的分支，因此origin/master中的内容不一定是git仓库里最新的内容；获取远程仓库最新的内容到origin/master需要靠git fetch命令。
2. 本地的远程分支origin/master和本地分支master也是两个不同的分支，还是得按分支合并那一套才能同步两个分支的内容。git push时一般是推送master的内容到git远程仓库的master分支。

先从远程仓库拉代码到本地，解决冲突+合并后，再推送到远程仓库的最标准的流程如下：

- 【git fetch origin master:master】 同步本地的origin/master分支和远程仓库的master分支；相当于origin/master有了一次最新提交

- 【git checkout master】切换到本地的master分支，【git merge origin/master】将origin/master的最新内容合并到master分支（这时候可能会有merge冲突。修改冲突之处后，再【git commit】一次冲突解决；此时master分支有origin/master中的新内容，也有本地更新的内容）
- 【git push origin master:master】将master分支上“在远程更新内容的基础上更新的本地内容”推送到远程仓库的master分支上

详细过程图解：

一、origin/master: ahead 2的意思是，距离上次把master 分支push上远程仓库，master已经有了两个新的提交节点，可以再次push了。**注意：不要手动在origin/master分支merge master，这是没有意义的。就算两个指针一致了，还是会显示ahead。只有push成功了才会重新同步，不显示ahead**

```
PS D:\code\zjy-learn\git-learn> git branch -vv
  dev    047006e 虽然没改但我解决冲突了
* master be194b3 [origin/master: ahead 2] 处理所有图片
```

git push后：

```
PS D:\code\zjy-learn\git-learn> git branch -vv
  dev    047006e 虽然没改但我解决冲突了
* master be194b3 [origin/master] 处理所有图片
```

二、origin/master:behind 2的意思是，远程仓库分两次改动了某些文件，并且分别fetch了两次到本地的origin/master分支上，但是此时的origin/master分支的内容并没有和master分支的内容合并，还需要【git checkout master】切换到本地master分支，【git merge origin/master】将origin/master的内容合并到master分支；master指针后移一个commit

```
PS D:\code\zjy-learn\git-learn> git fetch origin master
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 806 bytes | 115.00 KiB/s, done.
From https://github.com/ZengJiayi549/gitLearn
 * branch            master     -> FETCH_HEAD
   b06a548..7337fe9  master     -> origin/master
PS D:\code\zjy-learn\git-learn> git branch -vv
  dev    047006e 虽然没改但我解决冲突了
* master be194b3 [origin/master: behind 2] 处理所有图片
```

git checkout master, git merge origin/master后：

```
PS D:\code\zjy-learn\git-learn> git checkout master
Already on 'master'
M       problems&solu.md
Your branch is behind 'origin/master' by 2 commits, and can be fast-forwarded.
  (use "git pull" to update your local branch)
PS D:\code\zjy-learn\git-learn> git merge origin/master
Updating be194b3..7337fe9
Fast-forward
 ...266\351\200\240\346\210\220\347\232\204pull\345\206\262\347\252\201" | 2 ++
 1 file changed, 2 insertions(+)
 create mode 100644 "\346\265\213\350\257\225\350\277\234\347\250\213\344\270\2
6\210\220\347\232\204pull\345\206\262\347\252\201"
PS D:\code\zjy-learn\git-learn> git branch -vv
  dev    047006e 虽然没改但我解决冲突了
* master 7337fe9 [origin/master] Update 测试远程主机添加文件造成的pull冲突
```

三、如果在本地origin/master与本地master进行merge之前，master已经有过和远程主机改动同一文件的commit操作，会导致本地origin/master与本地master的merge产生冲突，需要手动修改（怎么修改，修不修改都不重要，重点是要再git commit一次）并【git commit】一遍来解决

```
PS D:\code\zjy-learn\git-learn> git branch -vv
  dev    047006e 虽然没改但我解决冲突了
* master 049430b [origin/master: ahead 2, behind 1] 本地master修改同一文件，还没push
```

改了2次，文件变更且fetch了1次，这种情况只要本地和远程修改了相同的文件（只要不是修改后的内容还一模一样）就会产生冲突

```
PS D:\code\zjy-learn\git-learn> git checkout master
Already on 'master'
Your branch and 'origin/master' have diverged,
and have 2 and 1 different commits each, respectively.
  (use "git pull" to merge the remote branch into yours)
PS D:\code\zjy-learn\git-learn> git merge origin/master
Auto-merging 测试远程主机添加文件造成的pull冲突
CONFLICT (content): Merge conflict in 测试远程主机添加文件造成的pull冲突
Automatic merge failed; fix conflicts and then commit the result.
```

手动修改冲突之处，再【git commit】就解决冲突并merge 了master和origin/master

```
PS D:\code\zjy-learn\git-learn> git commit -am "本地和远程版本都留下了，解决冲突的提交"
[master 0fd439a] 本地和远程版本都留下了，解决冲突的提交
```

此时master和origin/master已经在本地同步，远程也要同步的话，需要【git push】将master的内容推向远程仓库。这时由于远程主机没有再commit了，所以push是肯定不会有冲突的。至此，整个远程代码与本地代码同步->本地代码推送到远程仓库 加上解决冲突的过程就结束了。

```
PS D:\code\zjy-learn\git-learn> git add .
PS D:\code\zjy-learn\git-learn> git commit -m "解决图片问题，远程主机没再commit了，即将进行一个干净的push"
[master 011445d] 解决图片问题，远程主机没再commit了，即将进行一个干净的push
 4 files changed, 22 insertions(+), 5 deletions(-)
 create mode 100644 images/image-20230503163248408.png
 create mode 100644 images/image-20230503163605304.png
 create mode 100644 images/image-20230503163849093.png
PS D:\code\zjy-learn\git-learn> git branch -vv
  dev    047006e 虽然没改但我解决冲突了
* master 011445d [origin/master: ahead 4] 解决图片问题，远程主机没再commit了，即将进行一个干净的push
PS D:\code\zjy-learn\git-learn> git push origin master:master
Enumerating objects: 30, done.
Counting objects: 100% (28/28), done.
Delta compression using up to 8 threads
Compressing objects: 100% (22/22), done.
Writing objects: 100% (22/22), 325.42 KiB | 21.69 MiB/s, done.
Total 22 (delta 10), reused 0 (delta 0), pack-reused 0
To https://github.com/ZengJiayi549/gitLearn.git
   7456933..011445d  master -> master
```

插播一条 git push和git fetch最完整和标准的写法

【必看】git fetch 详解 - 简书 (jianshu.com)

1.git push :

repository 为远程主机地址，将本地指定分支推送到远程指定分支。

2.git push origin master

将本地的 master 分支推送到远程的 master 分支。

3.git push origin :master

删除远程 master 分支，当省略本地分支名时，效果等同于删除远程此分支。

4.git push origin

将当前分支推送到 origin 主机的对应分支。

5.git push

进行默认推送。

6.git push origin --delete

删除远程某个分支。

7.git push --all origin

将本地所有分支都推送到 origin 主机。

参考博客：

(92条消息) git push的详细使用_编程开发分享者的博客-CSDN博客

git创建本地与远程分支的同步与合并 - 掘金 (juejin.cn)git创建本地与远程分支的同步与合并 - 掘金 (juejin.cn)

Git极简教程(3)--branch级别的操作，合并分支 拉取最新代码git pull 包含(git fetch origin 和 git merge origin/master) - sunny123456 - 博客园 (cnblogs.com)

(92条消息) git push origin master和git push的区别_公孙元二的博客-CSDN博客

所以pull冲突本质就是merge冲突。那么push冲突是啥？和merge冲突有什么关系？ 如何解决?

如果没有先fetch+merge成功或pull成功后再push，只要远程仓库和本地要push上去的版本修改了同一文件，就会产生冲突，报错error: failed to push some refs to 'https://github.com/...

解决方法就是按上述流程先拉，再合并解决冲突，再推呗

也可以参考这一篇精简版的解决方法：

【必看】https://blog.csdn.net/qq_30152625/article/details/90404727

插播一条git log命令的参数：

(92条消息) git 之 git log命令原来还能这么玩Z One Dream的博客-CSDN博客

其他一定会用到的补充操作和理解：

撤销修改

(92条消息) Git撤销修改*git 撤销修改*路痴的兔子的博客-CSDN博客

(92条消息) git如何撤销工作区的修改*git撤销工作区修改*寅鳋的博客-CSDN博客

Git 之 reset --hard 回退/回滚到之前的版本代码后，后悔了，如何在恢复之后的版本的方法简单整理*git reset后怎么恢复*仙斛XAN的博客-CSDN博客

比较差异

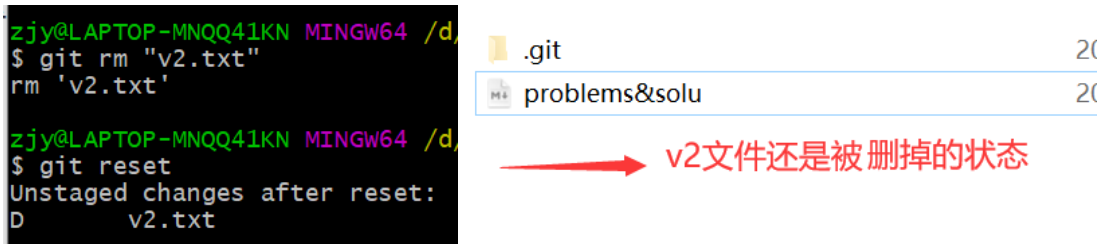
Git 基本命令 -- 你用过 git diff 吗？补习一下吧 - 知乎 (zhihu.com)

(92条消息) Git版本控制管理——diff_git diff 指定文件_止步听风的博客-CSDN博客

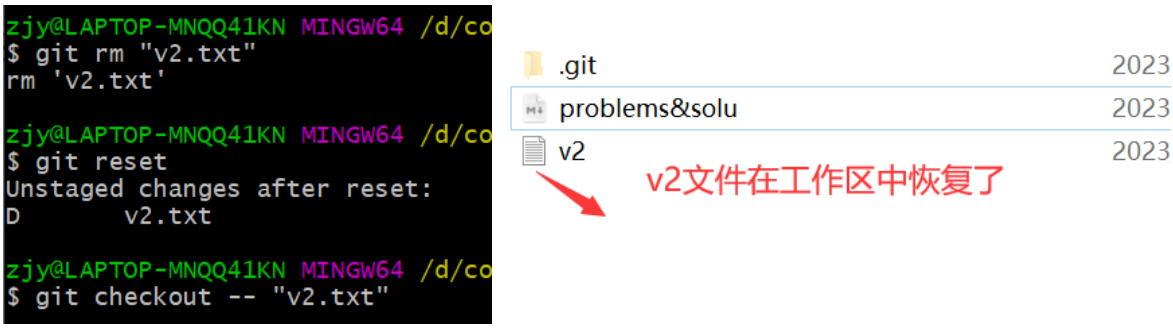
删除

git rm "filename.txt" 这个命令执行后是暂存区和工作区同步删除，没有commit到本地版本库；要撤回暂存区和工作区的删除操作分为两步：

- 撤回从本地删除操作被add到暂存区的操作 git reset



- 撤回本地删除操作 git checkout -- "filename.txt"



或者直接工作区和暂存区同时回退到版本库中上一次commit的状态 git reset --hard HEAD