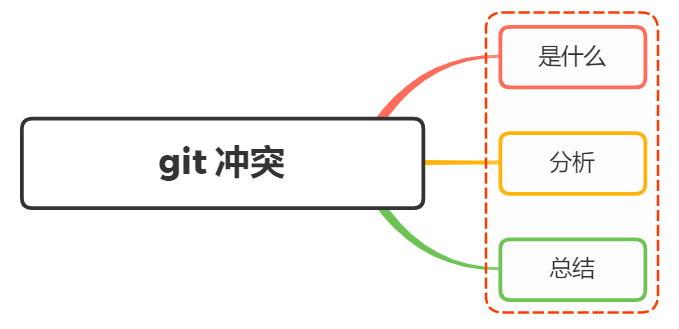
# 面试官：说说 git 发生冲突的场景？如何解决？



## 一、是什么

一般情况下，出现分支的场景有如下：

* 多个分支代码合并到一个分支时
* 多个分支向同一个远端分支推送

具体情况就是，多个分支修改了同一个文件（任何地方）或者多个分支修改了同一个文件的名称

如果两个分支中分别修改了不同文件中的部分，是不会产生冲突，直接合并即可

应用在命令中，就是push、pull、stash、rebase等命令下都有可能产生冲突情况，从本质上来讲，都是merge和patch（应用补丁）时产生冲突

## 二、分析

在本地主分值master创建一个a.txt文件，文件起始位置写上master commit，如下：



然后提交到仓库：

* git add a.txt
* git commit -m 'master first commit'

创建一个新的分支featurel1分支，并进行切换，如下：

git checkout -b featurel1

然后修改a.txt文件首行文字为 featurel commit，然后添加到暂存区，并开始进行提交到仓库：

* git add a.txt
* git commit -m 'featurel first change'

然后通过git checkout master切换到主分支，通过git merge进行合并，发现不会冲突

此时a.txt文件的内容变成featurel commit，没有出现冲突情况，这是因为git在内部发生了快速合并

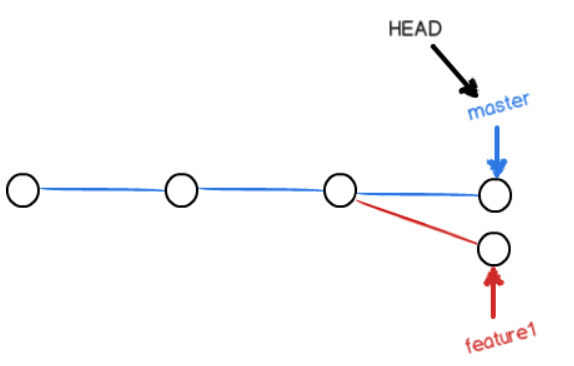
如果当前分支的每一个提交(commit)都已经存在另一个分支里了，git 就会执行一个“快速向前”(fast forward)操作

git 不创建任何新的提交(commit)，只是将当前分支指向合并进来的分支

如果此时切换到featurel分支，将文件的内容修改成featrue second commit，然后提交到本地仓库

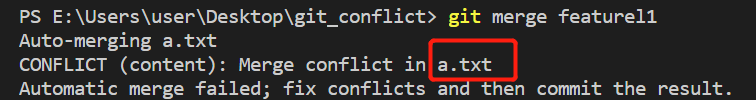
然后切换到主分支，如果此时在a.txt文件再次修改，修改成mastet second commit，然后再次提交到本地仓库

此时，master分支和feature1分支各自都分别有新的提交，变成了下图所示：



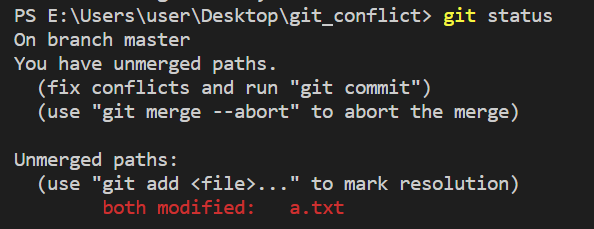
这种情况下，无法执行快速合并，只能试图把各自的修改合并起来，但这种合并就可能会有冲突

现在通过git merge featurel进行分支合并，如下所示：

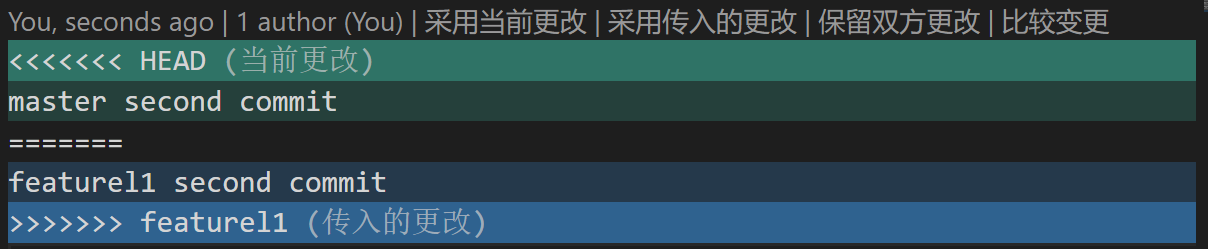


从冲突信息可以看到，a.txt发生冲突，必须手动解决冲突之后再提交

而git status同样可以告知我们冲突的文件：



打开a.txt文件，可以看到如下内容：



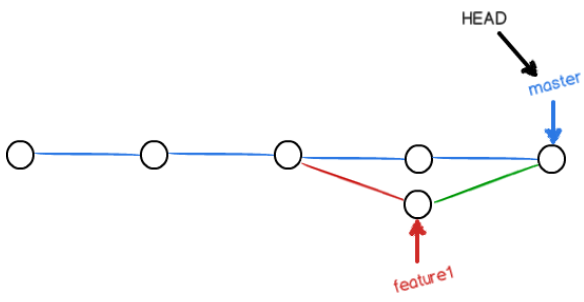
git用<<<<<<<，=======，>>>>>>>标记出不同分支的内容：

* <<<<<<< 和 ======= 之间的区域就是当前更改的内容
* ======= 和 >>>>>>> 之间的区域就是传入进来更改的内容

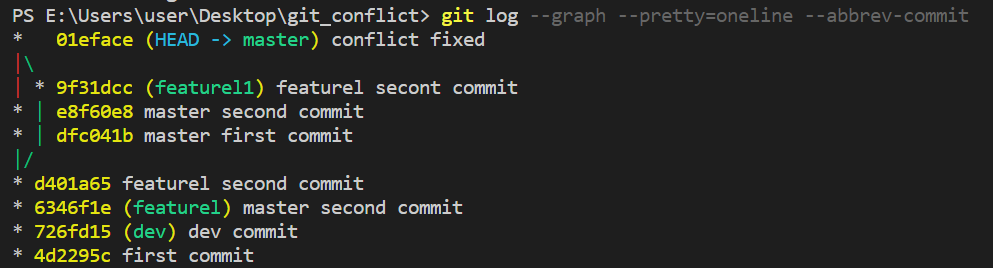
现在要做的事情就是将冲突的内容进行更改，对每个文件使用 git add 命令来将其标记为冲突已解决。 一旦暂存这些原本有冲突的文件，Git就会将它们标记为冲突已解决然后再提交：

* git add a.txt
* git commit -m "conflict fixed"

此时master分支和feature1分支变成了下图所示：



使用git log命令可以看到合并的信息：



## 三、总结

当Git无法自动合并分支时，就必须首先解决冲突，解决冲突后，再提交，合并完成

解决冲突就是把Git合并失败的文件手动编辑为我们希望的内容，再提交

## 参考文献

* https://www.liaoxuefeng.com/wiki/896043488029600/900004111093344