

Git提交错了，于是我把锅甩给了新来的baby

又是一遭悲惨的遭遇，git提交了一连串代码之后，发现提交错了。其实是把给老婆发的消息打到了comment里，然后还提交上去了。怎么办，这被看到岂不是要社死了。

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/4e466d9e66ac470a85303f02028b9def~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=657&h=190&s=11418&e=png&b=f8f8f8)

一连串的研究之后，找到了几个解决方案。接下来我们一起搞搞这种错误提交的弥补方案。其中最离谱的是第三个方案。哈哈。

赛前准备

这里模拟一下这个操作，毕竟不能直接看我们的代码记录。我们新建一个项目，新建一个文件，起名001。

![image.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/6b758be8c80148ab98e7e0b80a8773ea~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=579&h=267&s=14555&e=png&b=f7f7f7)

然后依次改为 002 003 004 005，每次都提交一次，在005的时候，执行异常提交。

最终我们得到一个005的文件

![image.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/36555d1d90e64575be692dc61a1858e~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=471&h=90&s=4605&e=png&b=ffffff)

gitee上看是这样的

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/3bde2b4d053d4487bcb14c974842d399~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=359&h=631&s=23112&e=png&b=ffffff)

对于我们来说，现在是想删除这个异常提交，不仅删除代码，还想删除记录

也就是说，期待的是，文件变为004，而且这个提交记录删除掉。

## 方案1 交互式 rebase

---

首先我们尝试一下 `git rebase -i HEAD~3`，这样会取出最后的三条提交记录供我们编辑。

![image.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/afa82ce620b14ff49c4f139c59937aab~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=722&h=410&s=50910&e=png&b=000000)

我们可以看到顶上有三条记录，这时候，我们删除这个异常的提交5

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/c8ca35a444c448a5a67921e56080e78c~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=289&h=74&s=5220&e=png&b=000000)

保存之后，会返回

```
```
git rebase -i HEAD~3
Successfully rebased and updated refs/heads/master.
```

这时候查看记录

![image.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/f79807692f2940c1b872f20fa2ab36e2~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=722&h=410&s=36313&e=png&b=000000)

异常提交已经没有了。

但是若是我们直接git push 会报错

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/aa35ff74b3564b69843b98071bcad66e~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=734&h=212&s=28440&e=png&b=000000)

告诉我们，我们当前的分支的版本是落后于远程分支的，不能提交。

这时候就需要`git push --force`这个命令，\*\*强制推送！！！\*\*

需要注意的是，\*\*强制推送会覆盖远程仓库中的历史记录\*\*，因此请确保你知道这个命令是个啥，并且有必要的话，需要通知团队其他成员协调好操作。

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/1641969c3c124b25be94d135e6f292a8~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=592&h=159&s=18882&e=png&b=000000)

可以看到，`git push --force` 是可以成功的，而且再看gitee的记录

![image.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/aaede3f2438d4f5b9d495b4c51b194e1~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=350&h=567&s=19722&e=png&b=ffffff)

异常提交5已经不见了。并且本地的文件已经变为了004

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/f3dea0138a2d4aec89f0c28d67a2ca12~tplv-k3u1fbpfcp-jj-

mark:3024:0:0:0:q75.awebp#?w=264&h=86&s=2611&e=png&b=fffffff)

其实在`git rebase -i HEAD~3`这个命令打开的交互框里是可以更改提交的顺序的，但是不能针对同一个文件的同一行，会冲突。

## 方案2 git reset

---

git reset 其实之前写文章讲过Git reset到底该如何使用，一文读懂系列 这次我们就直接为达目的，直接使用。

我们在上边的基础上，再提交一个异常提交5，使其恢复最初的情况。

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/9c0c55c91b8d4eaab04bb4ff617cc8bb~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=566&h=243&s=13718&e=png&b=f8f8f8)

然后gitee的情况：

![image.png](https://p1-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/9bcdea2c2a4a4625ba68b3bbfa4b2fd1~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=363&h=624&s=23483&e=png&b=fffffff)

这时候我们执行

```  
git reset --hard HEAD~1  
```

这个命令将删除最近的一个提交，包括提交所做的更改。请注意，这种方法可能会导致丢失未提交的更改，也就是说，本地写的没提交的代码就没了。所以请谨慎使用。

![image.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/1ceb884bf530434796980308ef173eeb~tplv-k3u1fbpfcp-jj-mark:3024:0:0:0:q75.awebp#?w=546&h=109&s=12119&e=png&b=000000)

执行之后，我们可以看到异常提交5不见了

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/55250421c6964344befdc0f661a2b5dc~tplv-k3u1fbpfcp-jj-mark:3024:0:0:q75.awebp#?w=722&h=410&s=36349&e=png&b=000000)

提交的时候也需要`git push --force`这个命令，强制推送！！！为啥每次都使用三个！！！呢，我只想告诉你，这个命令很恐怖，一定要慎之又慎。

这时候查看gitee记录

![image.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/6ce77e2d8e484ceea1879c0f107cb159~tplv-k3u1fbpfcp-jj-mark:3024:0:0:q75.awebp#?w=374&h=547&s=20151&e=png&b=ffffffff)

异常提交5没有了。

使用 git revert

---

还有小伴会说，为啥不用git revert呢，这不是git专门用来自回滚代码的吗？

我们恢复异常提交005，再试试

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/12f84cc7fec34e61977e7d991fc4c9e~tplv-k3u1fbpfcp-jj-mark:3024:0:0:q75.awebp#?w=1564&h=672&s=59237&e=png&b=ffffffff)

我们执行 `git revert f3d8db` 并且 push

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/1b695eb5d3a24f38a5cebe9b7439972b~tplv-k3u1fbpfcp-jj-mark:3024:0:0:q75.awebp#?w=573&h=311&s=36099&e=png&b=000000)

可以看到，文件是从005变为004了。但是从提交记录来看，不仅没有删除记录，还多了一条。其实，除非提交的注释特别社死，不然一般用的就是git revert，因为它不仅可以保存记录，还能确保版本是往前走的。

![image.png](https://p6-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/f79ece000f174494993eed07fb99985d~tplv-k3u1fbpfcp-jj-mark:3024:0:0:q75.awebp#?w=1592&h=759&s=73022&e=png&b=ffffff)

### 方案3 git filter-branch (谨慎使用)

---

查资料的时候，还看到一个这个命令，可以来一波骚的了。那既然提错了，把这锅甩给新人不就行了，哇咔咔咔咔。

```
...
git filter-branch --commit-filter '
  if git log --format="%B" -n 1 $GIT_COMMIT | grep -q "异常提交";
then
  GIT_AUTHOR_NAME="new baby";
  GIT_COMMITTER_NAME="new baby";
  git commit-tree "$@";
else
  git commit-tree "$@";
fi' -- --all
```

然后就是这样的

![image.png](https://p3-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/d30a56419df44d5492a59a4dada8277f~tplv-k3u1fbpfcp-jj-mark:3024:0:0:q75.awebp#?w=722&h=410&s=40881&e=png&b=000000)

![image.png](https://p9-juejin.byteimg.com/tos-cn-i-k3u1fbpfcp/617109b6c6024a5ea6254443fc97d78a~tplv-k3u1fbpfcp-jj-mark:3024:0:0:q75.awebp#?w=539&h=782&s=33522&e=png&b=ffffff)

可以看到名字变了。当然邮箱也是可以改的。哇咔咔，这异常不就与我没关系了么。。。但是，极其不建议这么瞎折腾哈。

这个命令会根据条件重写整个历史。操作之前备份一下吧，别折腾坏了。而且一定先和其他的小伙伴商量一下，尤其是新人哈。

在此，就研究完毕了。正常来说使用第一种或者第二种方案都是可以的。不怕挨打的话，第三种方案也行。

## git rebase 和 git reset 的区别

---

- \* `git rebase` 命令用于将一个分支的提交移动到另一个分支上，或者重新应用一系列的提交。它的主要作用是改变提交的基础，即重新设置提交的起点。
- \* `git reset` 命令用于修改当前分支的 HEAD 引用，或者用于撤销之前的提交操作。

也就是说`git rebase` 用于重新整理提交历史，而 `git reset` 用于调整当前分支的位置或撤销更改。关于这两个详细的使用，`git reset`已经写过了，有关`git rebase`的我会新开一篇文章，有关`将一个分支的提交移动到另一个分支上`这个操作虽不常用，但总有需要用到的时候。

原文链接: <https://juejin.cn/post/7365414174217355314>