**Git后悔药 - revert撤销（二）**

**4、一次移除某几次提交**

上面的演示中，只有两个提交需要 revert，我们可以一个个回退。但如果有几十个呢？一个个回退肯定效率太低而且容易出错。

（接上面示例）

我们可以使用以下方法进行批量回退：

**（1）git revert移除某几次提交的修改**

命令：git revert <commit6> <commit5> <commit4> …

|  |
| --- |
| 注意：提交的顺序是从最近的提交开始往前写。 |

示例：

|  |
| --- |
| Bash # 1.查看版本库历史提交记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/revert\_test (master) $ git log --oneline 30f7626 (HEAD -> master) Revert "第6次提交，revert 撤销第3次提交" d0c8e48 Revert "第5次提交，revert 撤销第4次提交" c04b29c 第4次提交，新增内容：revert test v4 fd819dc 第3次提交，新增内容：revert test v3 c71ae3c 第2次提交，新增内容：revert test v2 557f7c3 第1次提交，新增readme.txt文件  # 2.把第5、6次提交移除 # 先写提交5ID，在写提交6ID L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/revert\_test (master) $ git revert d0c8e48 30f7626 Auto-merging readme.txt CONFLICT (content): Merge conflict in readme.txt error: could not revert d0c8e48... Revert "第5次提交，revert 撤销第4次提交" hint: after resolving the conflicts, mark the corrected paths hint: with 'git add <paths>' or 'git rm <paths>' hint: and commit the result with 'git commit' |

说明：

* Auto-merging readme.txt：自动合并readme.txt文件出现问题。
* CONFLICT (content): Merge conflict in readme.txt：冲突（内容）：在readme.txt中合并冲突。
* error: could not revert d0c8e48... Revert "第5次提交，revert 撤销第4次提交"：无法还原d0c8e48提交。
* hint: after resolving the conflicts, mark the corrected paths：解决冲突后，把冲突文件提交到暂存区。
* hint: with 'git add <paths>' or 'git rm <paths>'：可使用“ git add <路径>”或“ git rm <路径>”方法，添加冲突文件到暂存区，或者删除文件。
* hint: and commit the result with 'git commit'：提示：并使用'git commit'提交结果。

我们应该先写commit-6，在写commit-5。

|  |
| --- |
| Bash # 1.撤销第5、6次提交 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/revert\_test (master) $ git revert 30f7626 d0c8e48 [master 31341a4] Revert "Revert "第7次提交，revert 撤销第5、6次提交""  1 file changed, 1 insertion(+) [master 509c208] Revert "Revert "第8次提交，revert 撤销第5、6次提交""  1 file changed, 1 insertion(+)  # 2.查看版本库历史提交记录 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/revert\_test (master) $ git log --oneline 509c208 (HEAD -> master) Revert "Revert "第8次提交，revert 撤销第5、6次提交"" 31341a4 Revert "Revert "第7次提交，revert 撤销第5、6次提交"" 30f7626 Revert "第6次提交，revert 撤销第3次提交" d0c8e48 Revert "第5次提交，revert 撤销第4次提交" c04b29c 第4次提交，新增内容：revert test v4 fd819dc 第3次提交，新增内容：revert test v3 c71ae3c 第2次提交，新增内容：revert test v2 557f7c3 第1次提交，新增readme.txt文件  # 3.查看readme.txt文件内容，又回到V4版本了 L@DESKTOP-T2AI2SU MINGW64 /j/git-repository/revert\_test (master) $ cat readme.txt revert test v1 revert test v2 revert test v3 revert test v4 |

我们可以看到，每撤销一个commit提交，都会生成一个新的提交。上面撤销了两个commit，就让你编写两次提交说明信息。

这时，错误的提交 commit-5 和 commit-6 依然保留，将来进行甩锅的时候也有依可循。而且，这样操作的话 HEAD 指针是往后移动的，可以直接使用 git push 命令推送到远程仓库里，**而这种做法，正是企业所鼓励的**。

**（2）git revert 移除某几次连续的提交的修改**

命令：git revert <commit新>...<commit旧>

（这里就不做演示了，同上）

**注意：**

* 含头不含尾，也就是撤销的提交，包含<commit新>，不包含<commit旧>。
* 提交的顺序也是，最新的commit开头，然后往前到比较远的commit提交。
* 移除过程中如果有冲突：
* 不知道某次提交的修改内容，可以使用 **git show <commit>** 查看修改内容。
* Git冲突的时候会提示当前要还原的commit的id是多少，如上error: could not revert d0c8e48... Revert "第5次提交，revert 撤销第4次提交"。
* 过程中移除了几次提交的修改，也会相应的生成几次revert commit记录。

**5、revert命令常用参数**

**（1）-n参数**

作用：git revert 命令移除某几次的提交的修改，但是不要使用还原的更改创建任何提交，还原只是修改工作树和索引。简写-n，全称--no-commit。

命令：git revert -n <commit新>...<commit旧>

**（2）-m参数**

作用：git revert 移除merge（合并）的修改内容。

简写：-m parent-number ，全称：--mainline parent-number。

命令：git revert [-m parent-number] <commit>。

说明：

如果我们需要还原的提交的是merge的操作，直接使用 git revert <commit> 的方式是不行的，会提示我们：

|  |
| --- |
| Bash error: commit ... is a merge but no -m option was given. fatal: revert failed |

因为merge操作有两个分支，而revert不知道要还原哪个分支的提交，就没法进行还原\移除那些提交记录的修改的操作了，所以我们用-m 1来告诉git revert命令哪个是主线，哪个是非主线。（选择主线就还原非主线，选择非主线就还原主线）

另外，这个主线是1还是2是从哪里看出来的呢？

我们可以使用 git show 命令查看，如果是普通的commit提交，通常会显示这个commit的id、作者、日期、备注、更改文件内容等；如果是merge的记录，则会显示merge 的commit-id，备注、日期和merge的主线和合并过来的id。

例如：

|  |
| --- |
| Bash commit 63374e93eebd2b86882e5a4bb75dcd9d0e334b15 Merge: bc64f61 72032b7 Author: unknown [test@163.com](mailto:test@163.com) Date: Sun Jul 19 11:34:21 2020 +0800    Merge branch ‘testcopy2’ into testcopy |

如上第二行，Merge: bc64f61 72032b7，bc64f61 就是主线的id，编号就是1；72032b7就是非主线的id，编号就是2。 此时，我们要把合并过来的分支的修改给还原掉，那么就可以指定1为主线，还原掉2的提交了。

**6、git revert和git reset命令的区别**

1. git revert命令是用一次新的commit来回滚之前的commit，git reset命令是直接回滚到指定的commit。
2. 即：git reset 命令是把HEAD向历史版本移动，而git revert命令是把HEAD继续指向新的commit。
3. 在回滚这一操作上看，效果差不多。但是在日后继续merge（合并）以前的老版本时有区别。
4. 因为git revert命令是用一次逆向的commit“中和”之前的提交，因此日后合并老的branch（分支）时，导致这部分改变不会再次出现。
5. 但是git reset命令是直接在某个分支上，回退到指定的commit，后边的提交不会在版本库的历史中（只能使用reflog查看）。因而和老的branch再次merge时，这些被回滚的commit应该还会被引入。

通过以上对比可以发现，git reset 命令与 git revert 命令最大的差别就在于，git reset 命令会失去后面的提交，而 git revert 命令是通过反做的方式重新创建一个新的提交，而保留原有的提交。在企业里，应尽量使用 git revert 命令，能不用 git reset 命令尽量不用。

|  |
| --- |
| 参考：   * [http://blog.itpub.net/31524109/viewspace-2653118/](https://links.jianshu.com/go?to=http%3A%2F%2Fblog.itpub.net%2F31524109%2Fviewspace-2653118%2F) * [https://www.talktocomputer.site/blogs/83/](https://links.jianshu.com/go?to=https%3A%2F%2Fwww.talktocomputer.site%2Fblogs%2F83%2F) |

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/307e83845cd3 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。