



链滴

# 修改 commit 记录的常用方法

作者: [vcjmhg](#)

原文链接: <https://ld246.com/article/1590380060958>

来源网站: [链滴](#)

许可协议: [署名-相同方式共享 4.0 国际 \(CC BY-SA 4.0\)](#)



## 概述

在我们日常使用git的过程中难免会出现commit提交有问题的情况，因此我将自己在日常开发过程中用的修改commit的方法总结如下。

## 修改commit历史

我们假设这样一个场景，程序员小吴创建了一个文本如下所示：

[illegible]

在没有仔细检查的情况下进行了一次提交：

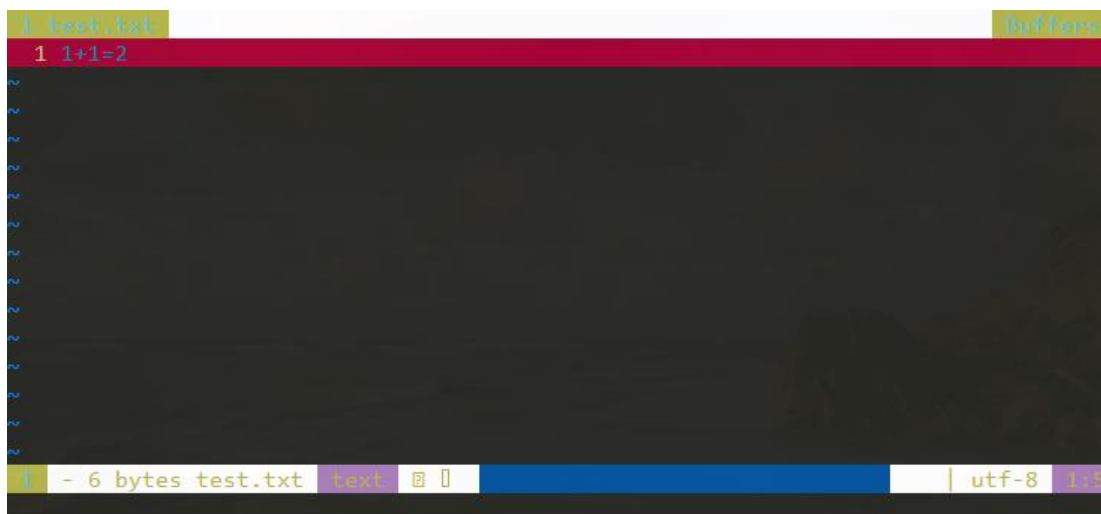
```
G:\gitWorkspace\test (master)
λ git add test.txt

G:\gitWorkspace\test (master)
λ git commit -"尝试第一次提交"
[master (root-commit) f1b61d0] 尝试第一次提交
1 file changed, 1 insertion(+)
create mode 100644 test.txt

G:\gitWorkspace\test (master)
λ git log --graph
* commit f1b61d053d3889ad4872894bad33ac00ad2e08e5 (HEAD -> master)
  Author: vcjmhg <13069648310@163.com>
  Date: Sun May 24 08:45:06 2020 +0800

    尝试第一次提交
```

然后此时小吴看到自己原来犯了一个低级错误写成了“1+1=1”，这太尴尬了，马上着手开始修改。



```
1 1+1=2
```

- 6 bytes test.txt text utf-8 1:5

修改完成之后，此时他就犯迷糊了：“如果直接提交是可以直接修改这个错误，但同时会有commit记录，万一被人发现这个小错误多尴尬。我吴某人一世英名可不能毁于一次commit呀”。此时小吴开网上搜集资料去解决这个问题，他很容易找到了第一种解决方法。

## 在commit时通过添加--amend参数进行修改

小吴思考之后执行了如下命令：

```
git add test.txt
git commit --amend -m "第二次正常提交"
```

```
G:\gitWorkspace\test (master)
λ git commit --amend -m "第二次正常提交"
[master 59082a4] 第二次正常提交
Date: Sun May 24 08:45:06 2020 +0800
1 file changed, 1 insertion(+)
create mode 100644 test.txt

G:\gitWorkspace\test (master)
λ git log --graph
* commit 59082a4a6438f17833d024fd2b0bea635dcd595e (HEAD -> master)
  Author: vcjmhg <13069648310@163.com>
  Date: Sun May 24 08:45:06 2020 +0800

    第二次正常提交
```

此时他通过`git log --graph`命令查看提交历史，果然第一次提交历史没有了。小吴很开心，继续很happy的写代码，但是写了一段时间之后，发现其实添加空格那次commit操作记录是没必要存在的。如还是刚才的方法显然没法改此次commit记录。

```
G:\gitWorkspace\test (master)
λ git log --graph
* commit deb6d0450e00089325fe071b62ed62c1cc03c0a0 (HEAD -> master)
  Author: vcjmhg <13069648310@163.com>
  Date:   Mon May 25 11:45:02 2020 +0800

    增加1+2

* commit b6dae6396c9597d28760da49cd7019e48a03882c
  Author: vcjmhg <13069648310@163.com>
  Date:   Mon May 25 11:43:35 2020 +0800

    添加一个空格

* commit afd330e5e705023f2529131f9f3ea31029da74c1
  Author: vcjmhg <13069648310@163.com>
  Date:   Mon May 25 11:30:28 2020 +0800

    第二次正常提交
```

因此小吴又走上了艰苦卓越的查找资料之路，功夫不负有心人，小吴终于找到了他想要的解决方法。

## 通过git rebase -i 命令将多次提交合并成一次提交

因此，小吴执行了如下命令

```
# HEAD~2表示HEAD前两次的提交，该例子指的是从commit id为ba9bb257
# 因为该命令在更改是必须给出想要修改的最近一次提交的父提交作为参数
# 命令等价于
# git rebase -i ba9bb257
git rebase -i HEAD~2
```

执行结果如下：

下边注释部分给出的是修改的命令格式写法。

由于我们是不想要添加一个空格这种commit记录的因此我们需要通过drop策略来删除此次commit史。

修改后的内容如下：

```
pick b6dae63 添加一个空格
s deb6d04 增加1+22

# Rebase afd330e..deb6d04 onto afd330e (2 commands)
#
# Commands:
# p, pick <commit> = use commit
# r, reword <commit> = use commit, but edit the commit message
# e, edit <commit> = use commit, but stop for amending
# s, squash <commit> = use commit, but meld into previous commit
# f, fixup <commit> = like "squash", but discard this commit's log message
# x, exec <command> = run command (the rest of the line) using shell
# b, break = stop here (continue rebase later with 'git rebase --continue')
# d, drop <commit> = remove commit
# l, label <label> = label current HEAD with a name
# t, reset <label> = reset HEAD to a label

<space/test/.git/rebase-merge/git-rebase-todo[+] [unix] (11:56 25/05/2020)2,1 顶端
:wq
```



保存后弹出新的界面如下：

```
# This is a combination of 2 commits.
# This is the 1st commit message:

添加一个空格

# This is the commit message #2:

增加1+2

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Mon May 25 11:43:35 2020 +0800
#
# interactive rebase in progress; onto afd330e
# Last commands done (2 commands done):
G:/gitWorkspace/test/.git/COMMIT_EDITMSG [unix] (11:57 25/05/2020) 4,1 顶端
"G:/gitWorkspace/test/.git/COMMIT_EDITMSG" [unix] 24L, 613C
```

删除“添加空格这一行”，然后保留“增加1+2”这一行。

```
# This is a combination of 2 commits.
# This is the 1st commit message:

# This is the commit message #2:

增加1+2

# Please enter the commit message for your changes. Lines starting
# with '#' will be ignored, and an empty message aborts the commit.
#
# Date:      Mon May 25 11:43:35 2020 +0800
#
# interactive rebase in progress; onto afd330e
# Last commands done (2 commands done):
#   pick b6dae63 添加一个空格
G:/gitWorkspace/test/.git/COMMIT_EDITMSG[+] [unix] (11:57 25/05/2020) 4,0-1 顶端
:
```

保存退出。查看提交记录：

```
λ git log
commit 01dc1ad01c6ff0b510564d24a522fb2a8a9399fd (HEAD -> master)
Author: vcjmhg <13069648310@163.com>
Date:   Mon May 25 11:43:35 2020 +0800

    增加1+2

commit afd330e5e705023f2529131f9f3ea31029da74c1
Author: vcjmhg <13069648310@163.com>
Date:   Mon May 25 11:30:28 2020 +0800

    第二次正常提交
```

## 总结

本文主要总结了常用的修改commit记录的两种方法，当然可能不太全面。欢迎各位小伙伴补充，谢谢！！