【快速理解Git分支:变基】【3】

fanlehai (/u/cc7ad717b6b1) ＋关注
2016.11.09 16:54* 字数 386 阅读 7947 评论 5 喜欢 11
(/u/cc7ad717b6b1)

- 在 Git 中整合来自不同分支的修改主要有两种方法：merge以及 rebase。
- **变基：**提取分支版本中引入的补丁和修改，然后合并到其他分支中。
- 变基操作的实质是丢弃一些现有的提交，然后相应地新建一些内容一样但实际上不同的提交。
- **变基的准则： 不要对在你的仓库外有副本的分支执行变基。**

【问题1】：合并图1中experiment分支到master分支上。

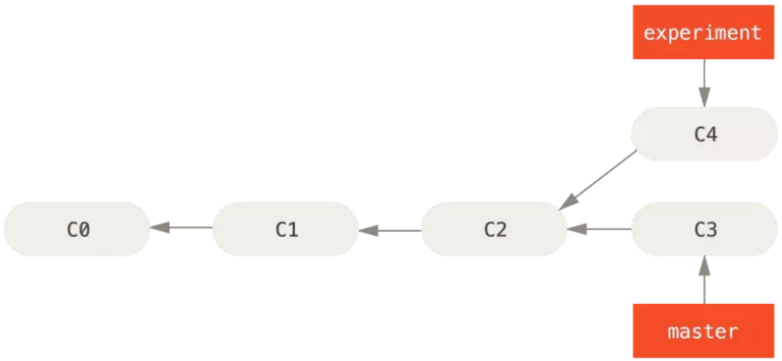


图1

【1】 通过merge合并分支：

```
# 合并命令如下：
$ git checkout master
$ git merge --no-ff -m "merge with no-ff" experiment
# 合并后如图2
```

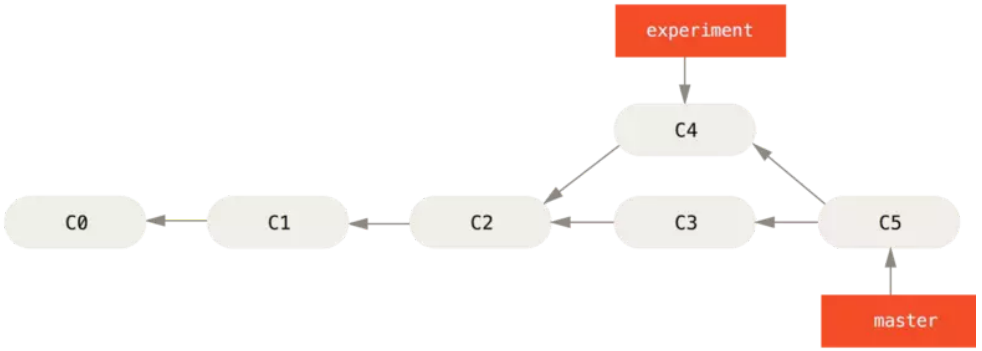



图2

> 【2】通过rebase合并分支：

```
#切换到experiment分支
  $ git checkout experiment
# 以master为基底分支，让experiment变基，如图3
  $ git rebase master
# experiment变基后回到master分支开始合并：如图4
  $ git checkout master
  $ git merge experiment
```

```
【变基原理：】
1.首先找到这两个分支（即当前分支 experiment、变基操作的目标基底分支 master）的最近共同祖先 C2；
2.然后对比当前分支相对于该祖先的历次提交，提取相应的修改并存为临时文件；
3.然后将当前分支指向目标基底 C3；
4.最后以此将之前另存为临时文件的修改依序应用到master分支
```

图3

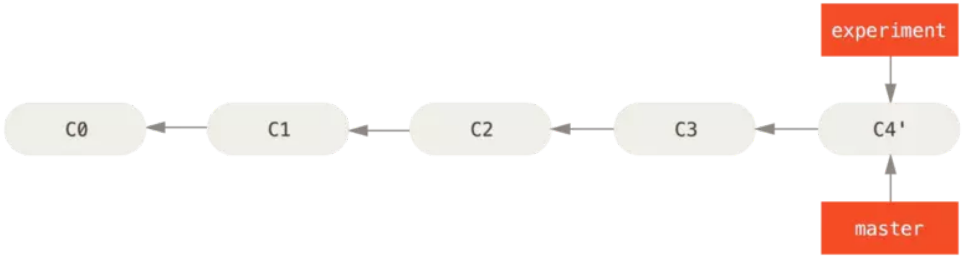


图4

【问题2】：合并图5中client分支到master分支上。

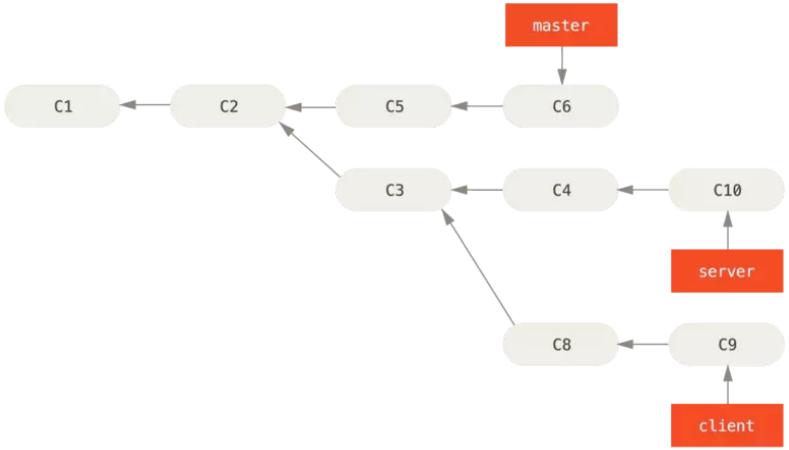


图5

```
【1】对client分支进行变基
    $ git rebase --onto master server client
#上面命令意思是:找出client和server的共同祖先之后所有client分支的修改，然后把它们在 master 分支上重
```

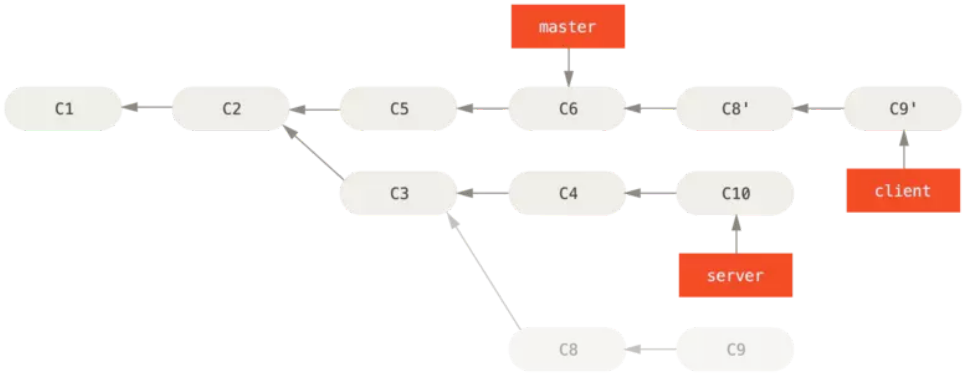



图6

```
【2】合并变基后client分支到master分支上，如图7
    $ git checkout master
    $ git merge client
```

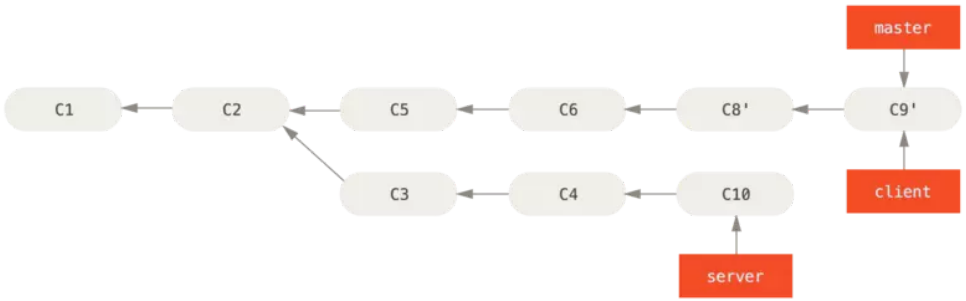


图7

> **【问题3】**：合并图7中server分支到master分支上。

```
1.对server分支进行变基（下面命令无需切换到server分支）图8
  $ git rebase master server
2.切换到master分支，开始合并
  $ git checkout master
  $ git merge server
3.删除无用分支
  $ git branch -d client
  $ git branch -d server
最终得到图9
```

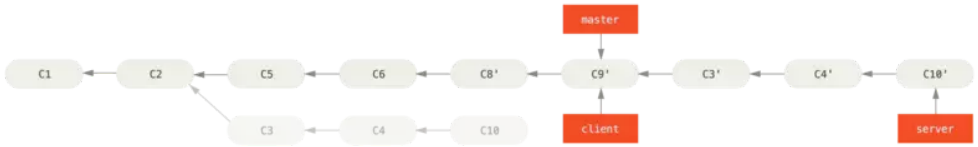



图8

图9

【问题4】：图10上面是远程仓库分区情况（这里对C4,C6分支执行了分基操作），下面是本地分支情况，此时本地代码想推送怎么处理？

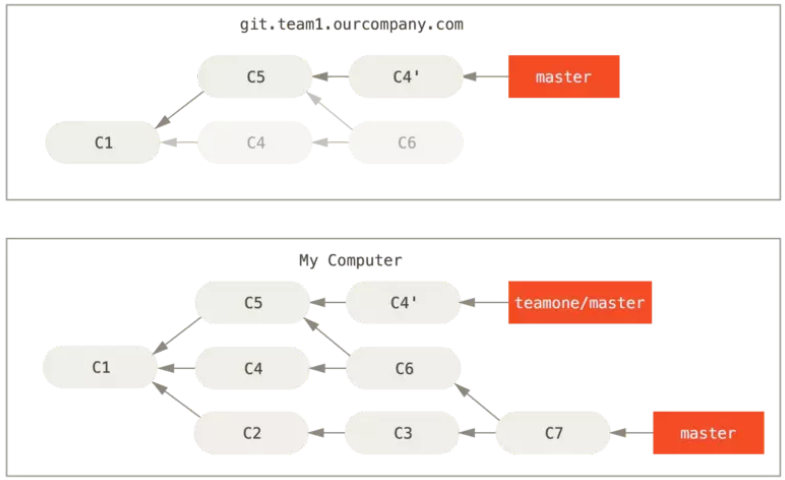


图10

【1】问题分析：正常操作是git pull拉取远程仓库最新代码，此操作会合并远程和本地代码，如图11，如果在推送此分支到远程，那么远程上对(C4,C6)执行的分基操作又被找回来了，这样会很混乱。

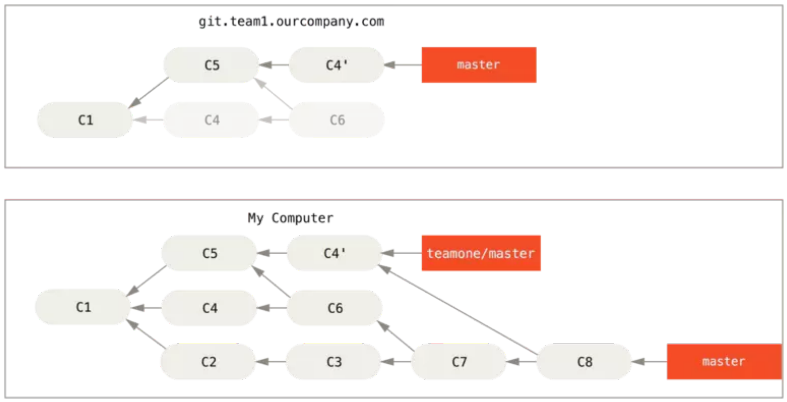
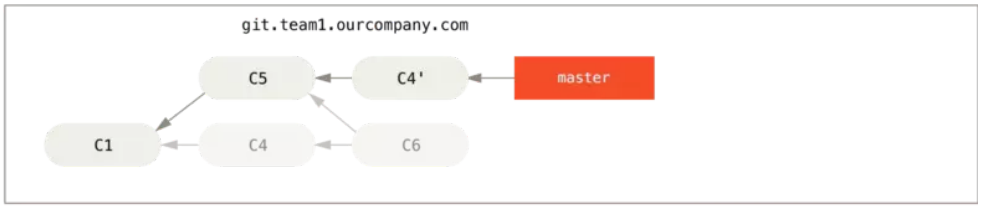


图11

> 【2】解决方案：用分基解决分基,不用git pull命令，使用下面命令：

```
法一：
  $ git pull --rebase
法二：
  $ git fetch
  $ git rebase teamone/master
上面两个方法等价
此命令执行操作如下：
1.检查哪些提交是我们的分支上独有的（C2，C3，C4，C6，C7）
2.检查其中哪些提交不是合并操作的结果（C2，C3，C4）
3.检查哪些提交在对方覆盖更新时并没有被纳入目标分支（只有 C2 和 C3，因为 C4 其实就是 C4'）
4.把查到的这些提交应用在 teamone/master上面
```



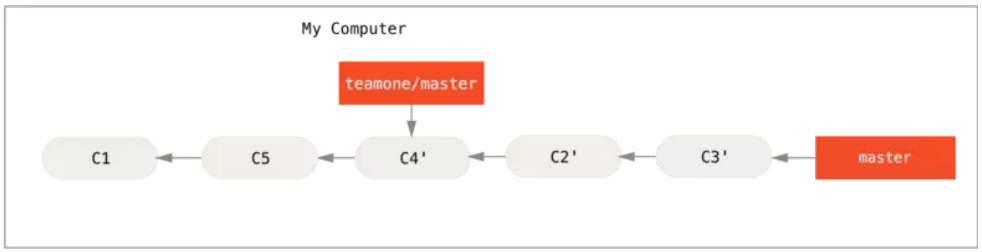


图12

赠人玫瑰手有余香！

赞赏支持