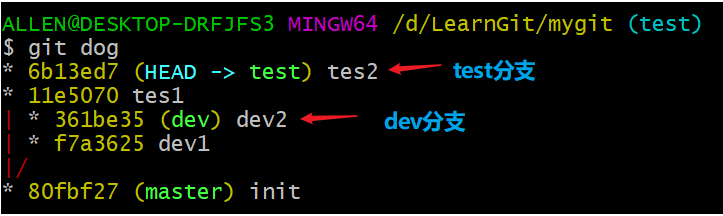
**Git操作 - rebase实战**

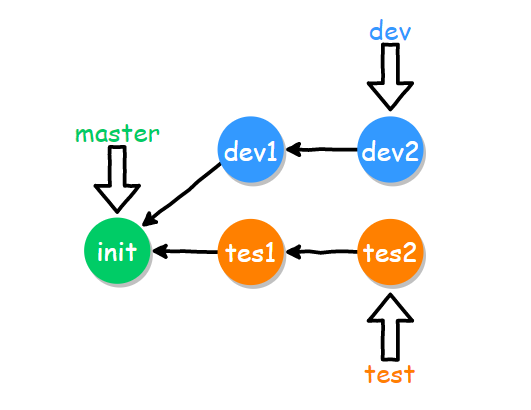
**实战一**

**使用git rebase命令，将多分个分支合并为一个分支。**

为了演示，额外创建两个分支dev和test，分别在两个分支上进行两次提交:



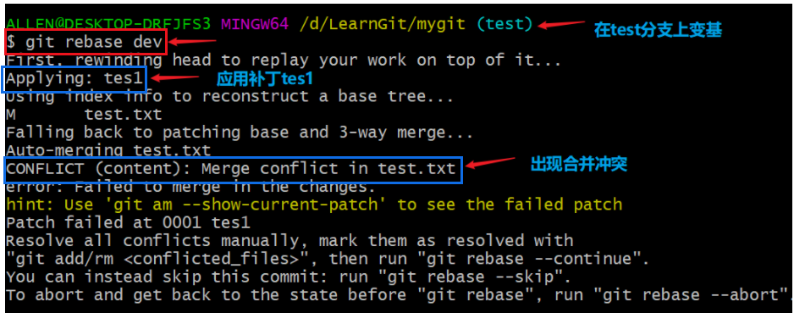
它们有一个共同的父提交节点init，此时本地仓库的状态如下图：



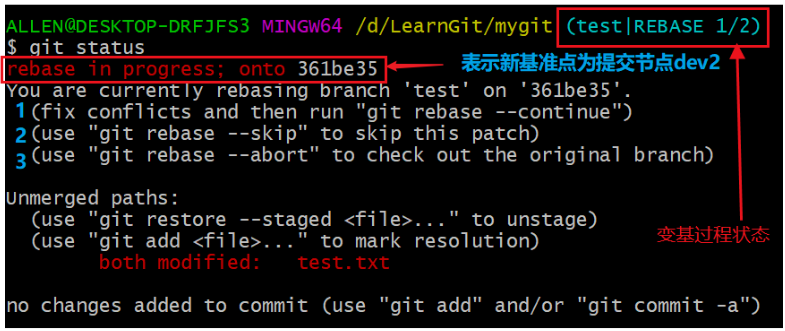
* 我们要对test分支进行变基，从而**合并到**dev分支上，所以需要先切换到test分支上，这与merge操作是相反的；
* 随后在test分支上执行git rebase dev命令，对该分支进行变基。
* 该指令翻译过来就是：test 分支要变更基准点，使用 dev 分支指向的提交，作为我新的基准点。

**1、变基分支，解决冲突**

**（1）首先，将test分支上的提交（补丁）tes1，应用到新基准点 dev2尾部，出现了合并冲突：**



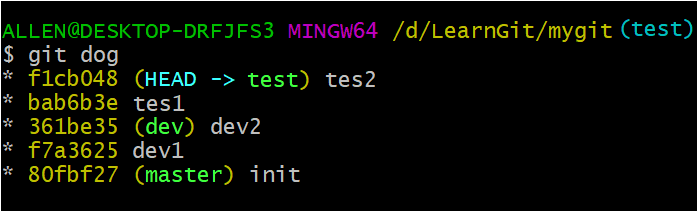
查看状态，发现test分支变基过程中的新基准点，正是dev分支指向的提交361be35，即提交节点dev2：



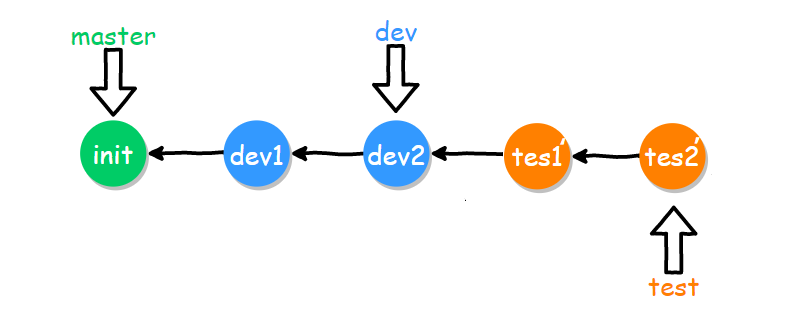
如图所示，此时有三个选项：

**选项一**：git rebase --continue：解决冲突，手动合并后，继续变基；

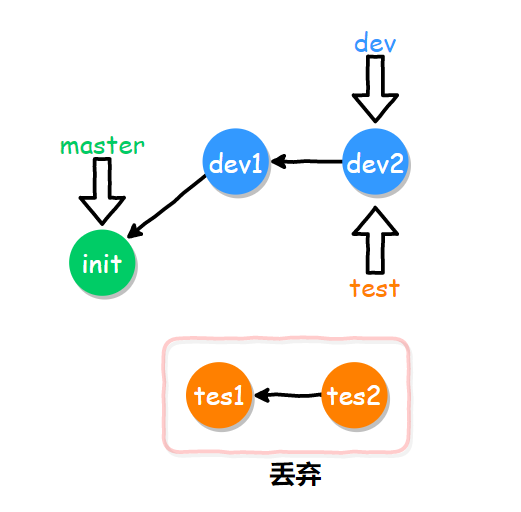
如下图：



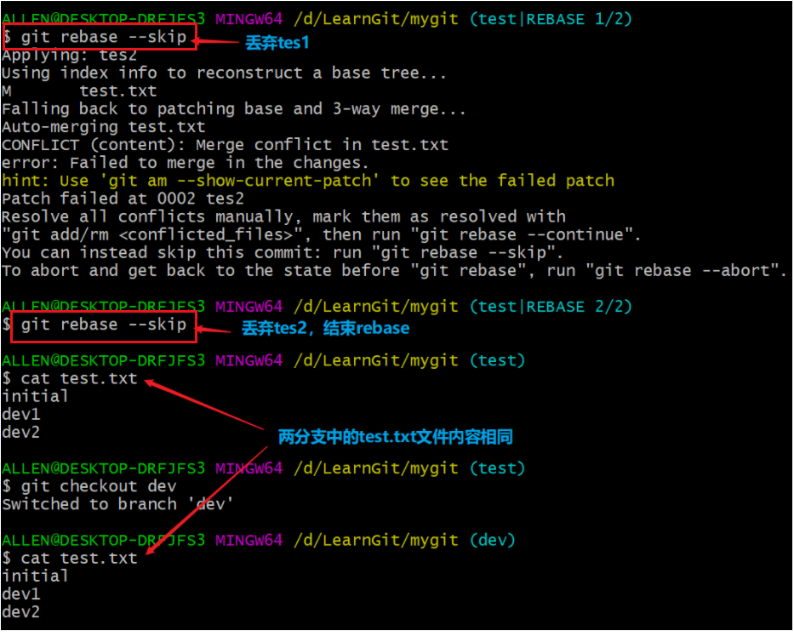
分支的状态如下所示：



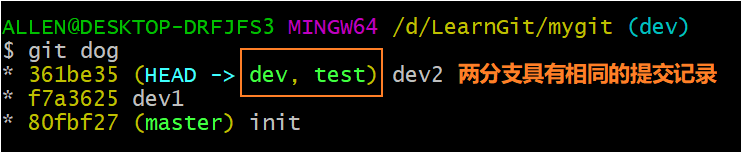
**选项二**：git rebase --skip：表示丢弃当前test分支的补丁，如果一直执行该选项，变基完成后，两分支的状态如下所示：



即此时test分支与dev分支上具有相同的文件：



**并且test分支上的提交记录被改变为了dev分支上的提交记录**。（test.txt文件的内容也被改变了）



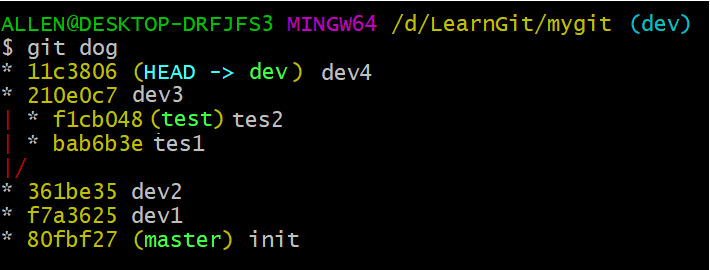
这就是一直执行选项git rebase --skip，丢弃全部test分支补丁的结果：

**选项三**：git rebase --abort：表示终止rebase操作，恢复到操作前；

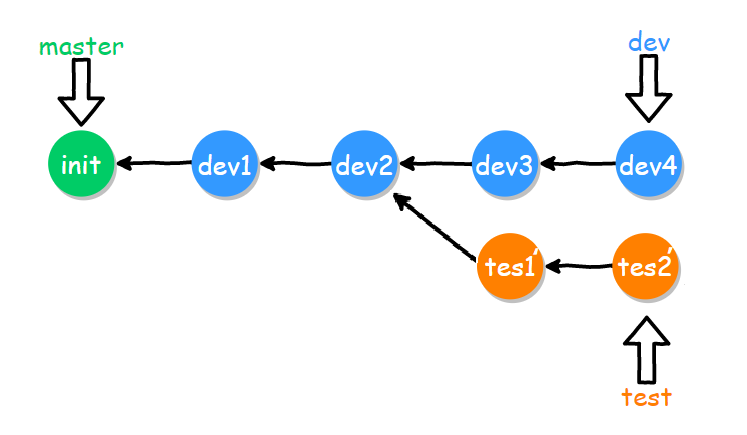
（我们选择选项一）

**2、生成分支，再次进行合并**

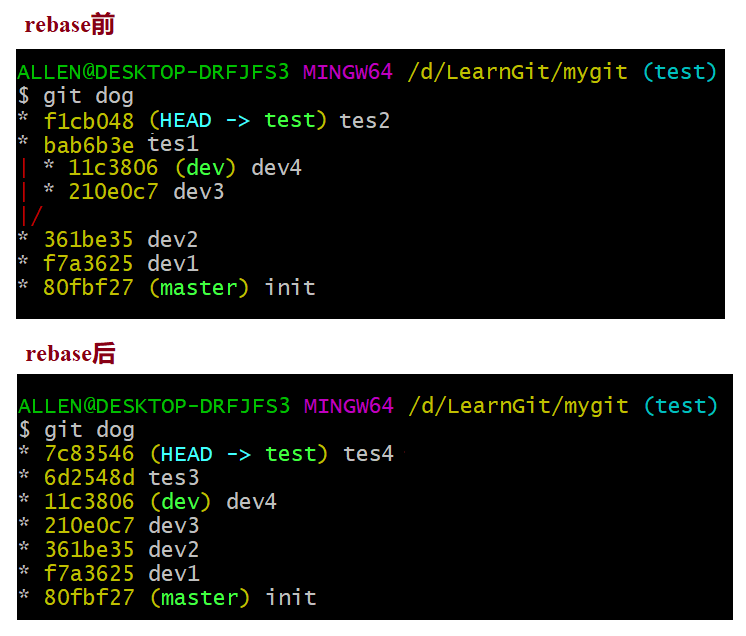
**（1）切换到dev分支上进行操作，在dev分支上新增两次提交dev3和dev4。**



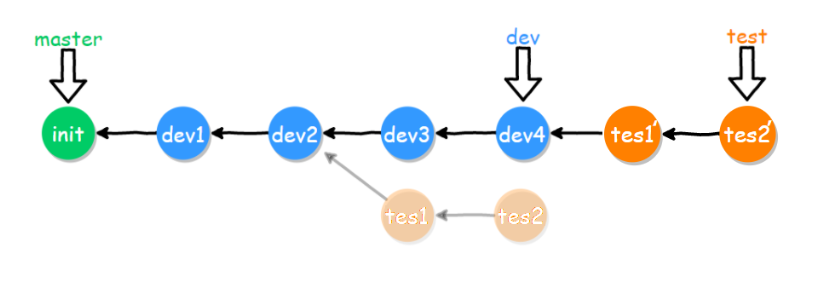
此时分支的状态如下所示：



**（2）执行命令git rebase dev，变基test分支到dev分支上。**



此时分支的状态如下所示：



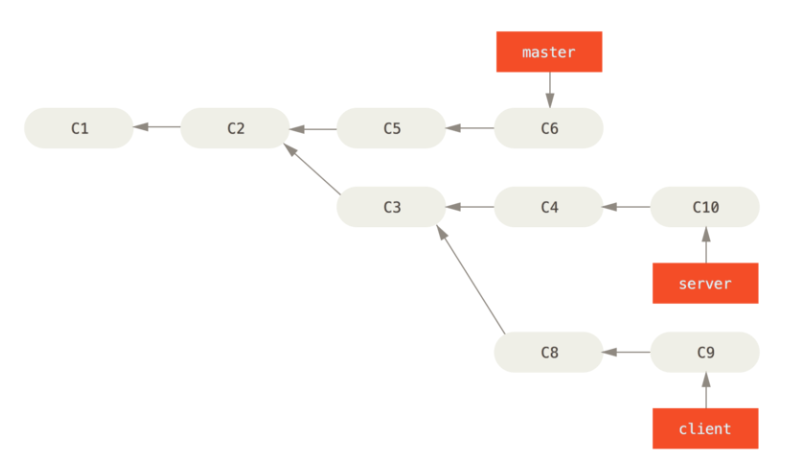
细心的你可能已经发现了，rebase与cherry-pick十分类似。只不过cherry-pick不会修改分支提交记录，而rebase会。

**实战二**

**1、情况说明**

在对两个分支进行变基时，所生成的“重放”并不一定要在目标分支上应用，你也可以指定另外的一个分支进行应用。

就像下图中，你创建了一个主题分支 server，为服务端添加了一些功能，提交了 C3 和 C4。 然后从 C3 上创建了分支 client，为客户端添加了一些功能，提交了 C8 和 C9。 最后，你回到 server 分支，又提交了 C10。



**2、合并分支一**

假设你希望将 client 分支中的修改，合并到主分支并发布。

但暂时并不想合并 server 中的修改， 因为它们还需要经过更全面的测试。

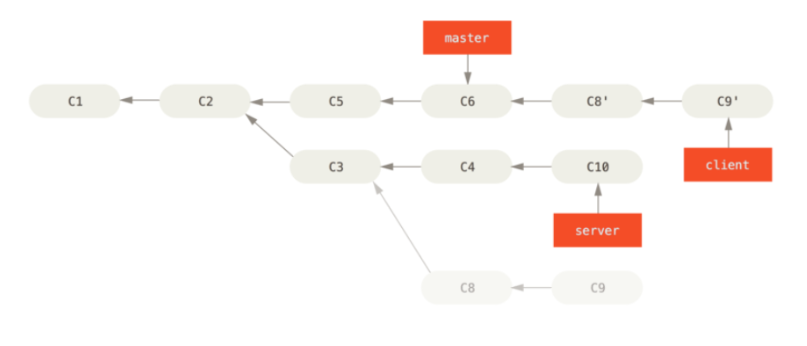
这时，你就可以使用 git rebase 命令的 --onto 选项， 选中在 client 分支里但不在 server 分支里的修改（即 C8 和 C9 提交），将它们在 master 分支上重放。

执行如下命令：

|  |
| --- |
| Bash $ git rebase --onto master server client |

以上命令的意思是：“取出 client 分支，找出它从 server 分支分歧之后的补丁， 然后把这些补丁在 master 分支上重放一遍，让 client 分支看起来像直接基于 master 分支修改一样”。这

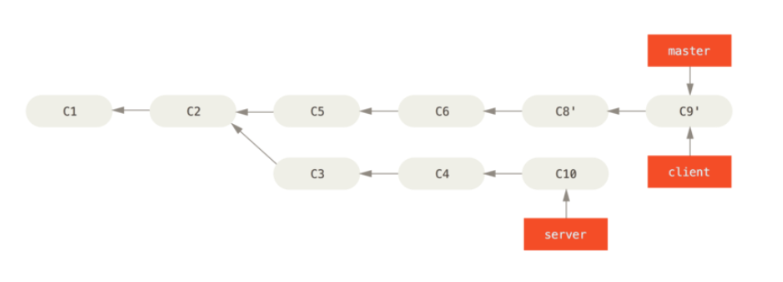
理解起来有一点复杂，不过效果非常酷，如下图：



现在可以快进合并 master 分支了。（如图 快进合并 master 分支，使之包含来自 client 分支的修改）：

|  |
| --- |
| Bash # 切换到master分支 $ git checkout master  # 把client分支merge到master分支 $ git merge client |

命令执行完，分支状态如下图所示：



**3、合并分支二**

接下来你决定将 server 分支中的修改也整合到 master 分支上。

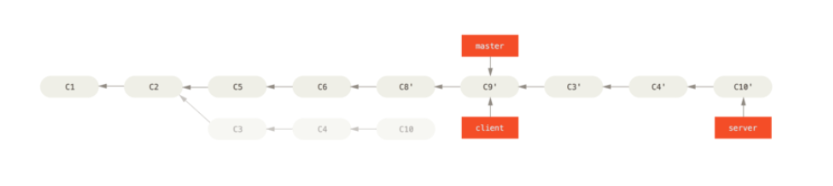
使用 git rebase 命令可以直接将 server 分支变基到目标分支上。

执行命令如下：

|  |
| --- |
| Bash # 此时在master分支上 $ git rebase master server |

说明：这样做能省去你先切换到 server 分支，再对其执行变基命令的多个步骤。

此时 server 分支中的代码，被“续”到了 master 分支的后面，如下图所示：



然后就可以把 server 分支，快进合并主分支 master 了：

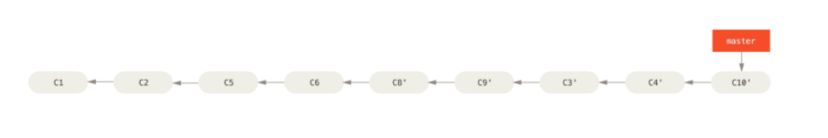
|  |
| --- |
| Bash $ git checkout master $ git merge server |

至此，client 分支和 server 分支中的修改，都已经整合到主分支里了。

你可以删除这两个分支。

|  |
| --- |
| Bash $ git branch -d client $ git branch -d server |

最终提交历史记录会变成如下图所示：



参考：

* [https://blog.csdn.net/qq\_43425487/article/details/105620270](https://links.jianshu.com/go?to=https%3A%2F%2Fblog.csdn.net%2Fqq_43425487%2Farticle%2Fdetails%2F105620270)
* [https://blog.csdn.net/weixin\_38937840/article/details/113250868](https://links.jianshu.com/go?to=https%3A%2F%2Fblog.csdn.net%2Fweixin_38937840%2Farticle%2Fdetails%2F113250868)

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/a794bbd24d78 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。