

< 2019年3月 >						
日	一	二	三	四	五	六
24	25	26	27	28	1	2
3	4	5	6	7	8	9
10	11	12	13	14	15	16
17	18	19	20	21	22	23
24	25	26	27	28	29	30
31	1	2	3	4	5	6



公告

昵称：sinojelly
园龄：9年1个月
粉丝：41
关注：1
[+加关注](#)



搜索

<input type="text"/>	找我看
<input type="text"/>	谷歌搜索



常用链接

[我的随笔](#)
[我的评论](#)
[我的参与](#)
[最新评论](#)
[我的标签](#)
[更多链接](#)

Git下的冲突解决

Posted on 2011-08-07 18:03 sinojelly 阅读(248745) 评论(1) [编辑](#) [收藏](#)

冲突的产生

很多命令都可能出现冲突，但从根本上来讲，都是merge 和 patch（应用补丁）时产生冲突。而rebase就是重新设置基准，然后应用补丁的过程，所以也会冲突。git pull会自动merge，repo sync会自动rebase，所以git pull和repo sync也会产生冲突。当然git rebase就更不用说了。

冲突的类型

逻辑冲突

git自动处理（合并/应用补丁）成功，但是逻辑上是有问题的。比如另外一个人修改了文件名，但我还使用老的文件名，这种情况下自动处理是能成功的，但实际上是有问题的。又比如，函数返回值含义变化，但我还使用老的含义，这种情况自动处理成功，但可能隐藏着重大BUG。这种问题，主要通过自动化测试来保障。所以最好是能够写出比较完备的自动化测试用例。这种冲突的解决，就是做一次BUG修正。不是真正解决git报告的冲突。

内容冲突

两个用户修改了同一个文件的同一块区域，git会报告内容冲突。我们常见的都是这种，后面的解决办法也主要针对这种冲突。

树冲突

文件名修改造成的冲突，称为树冲突。比如，a用户把文件改名为a.c，b用户把同一个文件改名为b.c，那么b将这两个commit合并时，会产生冲突。

```
$ git status
added by us:      b.c
both deleted:    origin-name.c
added by them:    a.c
```

如果最终确定用b.c，那么解决办法如下：



我的标签

总结(1)



随笔档案

- 2013年4月 (1)
- 2012年9月 (1)
- 2012年1月 (3)
- 2011年11月 (2)
- 2011年8月 (2)
- 2011年7月 (1)
- 2011年6月 (1)
- 2011年4月 (1)
- 2011年3月 (1)
- 2011年1月 (5)
- 2010年12月 (2)
- 2010年11月 (9)
- 2010年10月 (7)
- 2010年9月 (1)
- 2010年8月 (1)
- 2010年6月 (1)
- 2010年5月 (6)
- 2010年4月 (3)
- 2010年3月 (11)
- 2010年2月 (20)



最新评论

1. Re:CMake入门指南
Windows上安装CMake,并且使用,看上去实在是太痛苦了
--从此寂静无声
2. Re:CMake入门指南
源代码的Tests/Tutorial目录中找到这个手册对应的代码。。。这个代码哪里可以下载呢
--shirly650
3. Re:CMake入门指南

```
git rm a.c
git rm origin-name.c
git add b.c
git commit
```

执行前面两个git rm时，会告警“file-name : needs merge”，可以不必理会。

树冲突也可以用git mergetool来解决，但整个解决过程是在交互式问答中完成的，用d 删除不要的文件，用c保留需要的文件。最后执行git commit提交即可。

内容冲突的解决办法

发现冲突

一般来讲，出现冲突时都会有“CONFLICT”字样：

```
$ git pull
Auto-merging test.txt
CONFLICT (content): Merge conflict in test.txt
Automatic merge failed; fix conflicts and then commit the result.
```

但是，也有例外，repo sync的报错，可能并不是直接提示冲突，而是下面这样：

```
error: project mini/sample
```

注：无论是否存在冲突，只要本地修改不是基于服务器最新的，它都可能报告这个错误，解决方法都是一样。

这个时候，需要进入报错的项目（git库）目录，然后执行git rebase解决：

```
git rebase remote-branch-name
```

冲突解决的一般过程

merge/patch的冲突解决

先编辑冲突，然后git commit提交。

注：对于git来讲，编辑冲突跟平时的修改代码没什么差异。修改完成后，都是要把修改添加到缓存，然后commit。

rebase的冲突解决

rebase的冲突解决过程，就是解决每个应用补丁冲突的过程。

解决完一个补丁应用的冲突后，执行下面命令标记冲突已解决（也就是把修改内容加入缓存）：

```
git add -u
```

注：-u 表示把所有已track的文件的新的修改加入缓存，但不加入新的文件。

然后执行下面命令继续rebase：

```
git rebase --continue
```

有冲突继续解决，重复这这些步骤，直到rebase完成。

我在“使用步骤”那块没有生成makefile文件是啥原因呢，并且make的时候，出错“make不是内部命令或外部命令，也不是可执行程序或批处理文件”

--winner_be

4. Re:CMake入门指南

翻译的不错，英文好的话，看原文更好

--tuling56

5. Re:CMake入门指南

写的太全面了

--思维与睿智

阅读排行榜

1. Git下的冲突解决(248745)
2. 简明教程：用Intel My Wifi技术把笔记本电脑改造成无线路由器(217065)
3. CMake入门指南(115951)
4. Chrome扩展程序开发调试简明教程(17303)
5. Windows平台ipod touch/iphone等共享笔记本无线上网设置大全(11600)

评论排行榜

1. 简明教程：用Intel My Wifi技术把笔记本电脑改造成无线路由器(16)
2. CMake入门指南(15)
3. APK编译时自动带上commit id版本标识(9)
4. Compilation problem with oprofile regarding libbfd on latest android-4.2.2_r1 branch?(7)
5. dropbox on ubuntu(6)

推荐排行榜

1. Git下的冲突解决(13)

如果中间遇到某个补丁不需要应用，可以用下面命令忽略：

```
git rebase --skip
```

如果想回到rebase执行之前的状态，可以执行：

```
git rebase --abort
```

注：rebase之后，不需要执行commit，也不存在新的修改需要提交，都是git自动完成。

编辑冲突的方法

直接编辑冲突文件

冲突产生后，文件系统中冲突了的文件（这里是test.txt）里面的内容会显示为类似下面这样：

```
a123
<<<<<< HEAD
b789
=====
b45678910
>>>>>> 6853e5ff961e684d3a6c02d4d06183b5ff330dcc
c
```

其中：冲突标记<<<<<<（7个<）与=====之间的内容是我的修改，=====与>>>>>>之间的内容是别人的修改。此时，还没有任何其它垃圾文件产生。

最简单的编辑冲突的办法，就是直接编辑冲突了的文件（test.txt），把冲突标记删掉，把冲突解决正确。

利用图形界面工具解决冲突

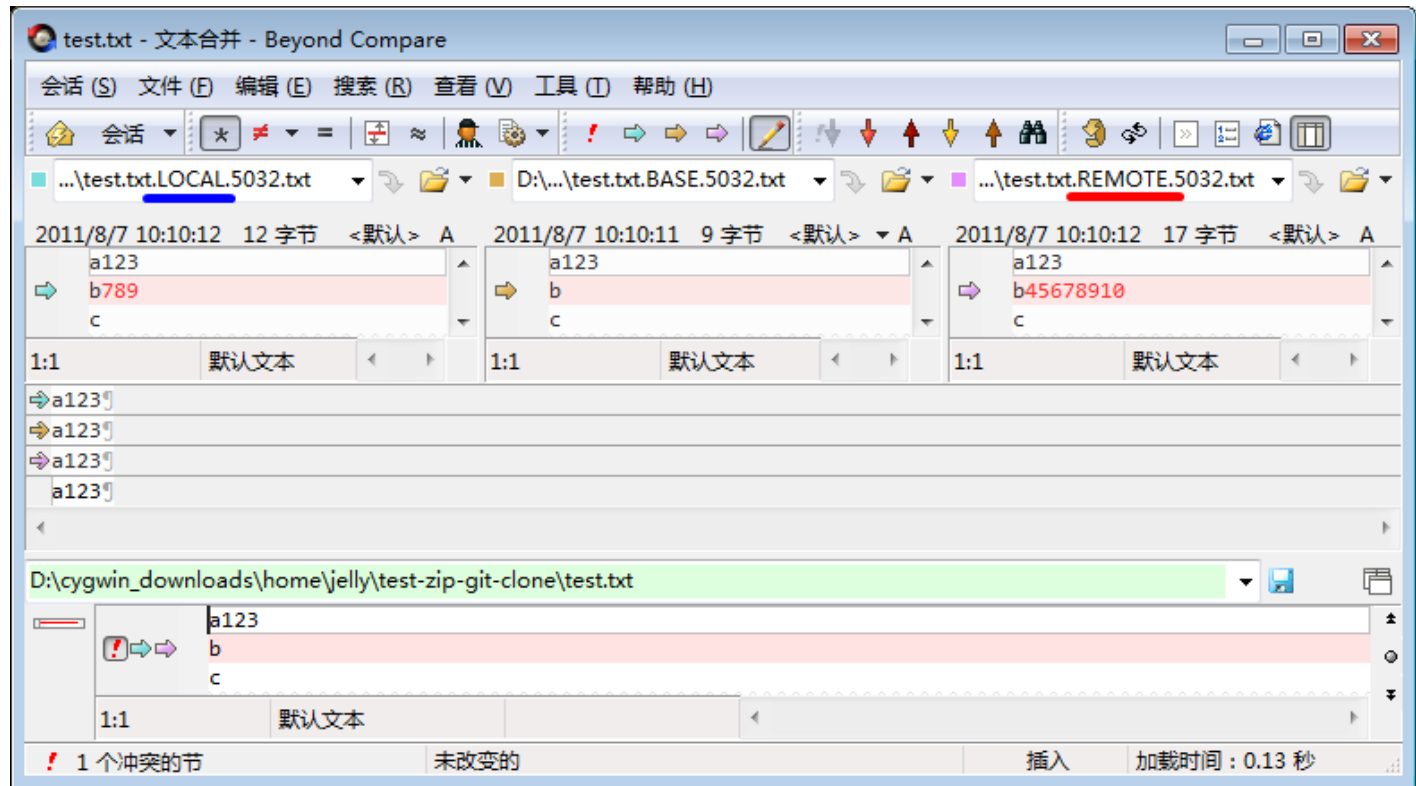
如果要解决的冲突很多，且比较复杂，图形界面的冲突解决工具就显得很重要了。

执行git mergetool用预先配置的Beyond Compare解决冲突：

```
$ git mergetool
Merging:
test.txt

Normal merge conflict for 'test.txt':
  {local}: modified
  {remote}: modified
```

2. CMake入门指南(12)
3. Chrome扩展程序开发调试简明教程(5)
4. Python脚本打包为exe文件(1)
5. 揭开PC-Lint9的神秘面纱(1)



上面三个窗口依次是“LOCAL”、“BASE”、“REMOTE”，它们只是提供解决冲突需要的信息，是无法编辑的。
下面一个窗口是合并后的结果，可以手动修改，也可以点击相应颜色的箭头选择“LOCAL”或者“REMOTE”。

在Beyond Compare中修改冲突保存后，冲突文件（test.txt）中的冲突标记就没有了，成了修改后的内容，一个文件的冲突编辑就完成了。

注意：

启动Beyond Compare之后，会自动生成几个包含大写字母名称、数字的辅助文件：

名称

- .git
- test.txt
- test.txt.BACKUP.5032.txt
- test.txt.BASE.5032.txt
- test.txt.LOCAL.5032.txt
- test.txt.REMOTE.5032.txt

关闭Beyond Compare时，这几个辅助文件都会自动删除，但同时会生成一个test.txt.orig的文件，内容是解决冲突前的冲突现场。
默认该.orig文件可能不会自动删除，需要手动删掉。

[好文要顶](#)[关注我](#)[收藏该文](#)[sinojelly](#)[关注 - 1](#)[粉丝 - 41](#)[+加关注](#)[« 上一篇 : Git Repo Gerrit方式下如何进行代码读权限控制](#)[» 下一篇 : Git下使用Beyond Compare作为比较和合并工具](#)

13

1

Feedback

#1楼

2014-10-10 14:40 by 乐乐龙王

谢谢楼主

[支持\(1\)](#) [反对\(0\)](#)[刷新评论](#) [刷新页面](#) [返回顶部](#)

注册用户登录后才能发表评论，请 [登录](#) 或 [注册](#)，[访问网站首页](#)。

【幸运】99%的人不知道我们有可以帮你薪资翻倍的秘笈！

【推荐】超50万C++/C#源码：大型实时仿真组态图形源码

【推荐】百度云“猪”你开年行大运，红包疯狂拿

【推荐】55K刚面完Java架构师岗，这些技术你必须掌握

相关博文：

- [git解决冲突](#)
- [Git冲突和解决冲突](#)
- [git解决冲突](#)
- [Git下解决冲突](#)
- [git 冲突解决](#)

最新新闻：

- 315晚会揭露7大黑料，电子烟、辣条、骚扰电话都上榜了
 - 51Talk去年净营收11.5亿元，同比增长35%，今年重点仍是城市下沉
 - 刘强东渴望一个CTO
 - 19年Edward J. McCluskey技术成就奖揭晓，周志华教授成唯一获奖者
 - 亚勤的毕业季 百度的新学期
- » [更多新闻...](#)

