

如何解决Git合并冲突

2017-03-31

背景

1. 如果你的工作目录的代码做了修改，但又尚未提交，此时你不小心做了 ``git pull`` 操作拉取远程仓库代码到本地时，往往会提示冲突。
2. 你在分支 ``dev`` 上完成了功能开发，然后 ``checkout master``，``git merge dev``，此时如果其他人不慎对 ``master`` 进行了修改，往往也会提示冲突。

遇到以上两个问题，你该如何解决呢？

针对 问题1 的解决办法

代码尚未完成

如果你当前的本地代码尚未完成，此时推荐使用 `stash` 命令来暂存修改，避免将未完成的功能代码 `commit` 到本地仓库，污染仓库。操作步骤如下：

`stash` 暂存当前的代码修改：

- 1 `$ git stash save '测试'`
- 2 `Saved working directory and index state On master: 测试`

检查:

- 1 `$ git status`
- 2 `On branch master`

```
3  nothing to commit, working tree clean
```

上面 `测试` 文字为你当前存储备注。完成后使用 `git status` 命令可以看到暂存后工作目录非常干净，这是因为：

Stashing takes the dirty state of your working directory – that is, your modified tracked files and staged changes – and saves it on a stack of unfinished changes that you can reapply at any time.

`git pull` 来去远程仓库并自动合并（当然如果你的仓库跟远程仓库还有冲突的话，那就是问题2了）

```
1  $ git pull
```

`stash pop` 重新还原暂存区的修改内容：

```
1  $ git stash pop
```

代码已经完成

如果你的代码已经完成并且确认没有错误，就可以先 `commit` 到本地仓库。再次 `pull` 拉取远程仓库时，如无冲突，Git 会自动产生一次 合并提交，这是因为 `pull` 的默认策略是 `fetch + merge`。如果本地仓库的 `commit` 一直不 `push` 到远程仓库的话，极端情况下，每次都可能产生一次 合并操作，这会造成 `graph` 非常复杂。个人建议直接 `rebase`，避免本地仓库无谓的合并节点：

```
1  $ git pull --rebase
```

针对 问题2 解决办法

手动修改冲突文件

当你执行 `git merge` 命令如下：

```
1 $ git merge dev
2 Auto-merging test2.txt
3 CONFLICT (content): Merge conflict in test2.txt
4 Auto-merging test.txt
5 CONFLICT (content): Merge conflict in test.txt
6 Automatic merge failed; fix conflicts and then commit the result.
```

那么你只需要打开相应的冲突文件进行手动代码修改操作。

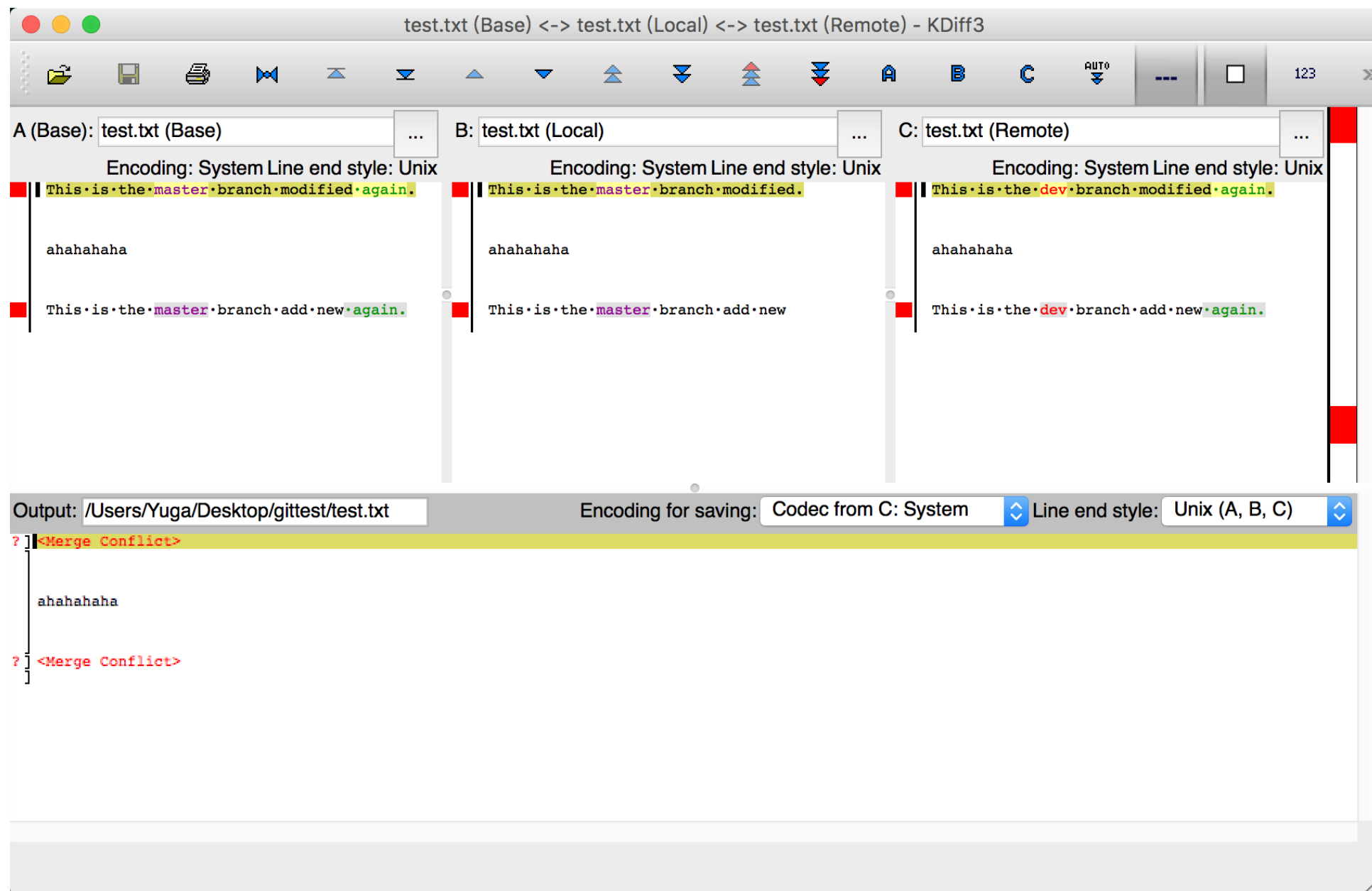
使用 git mergetool

当执行 `git merge` 命令，出现上面问题后，直接执行

```
1 $ git mergetool
2 This message is displayed because 'merge.tool' is not configured.
3 See 'git mergetool --tool-help' or 'git help config' for more details.
4 'git mergetool' will now attempt to use one of the following tools:
5 tortoisemerge emerge vimdiff opendiff
6 Merging:
7 test.txt
8
9 Normal merge conflict for 'test.txt':
10  {local}: modified file
11  {remote}: modified file
12 Hit return to start merge resolution tool (vimdiff):
```

一般 Mac 会让你选择相应的tool，你只需要输入相应的工具名称即可，前提是你的电脑已经安装，比如我的安装了 [kdiff3](#)，我只需要输入 kdiff3即可。

然后会自动启动kdiff3界面如下：



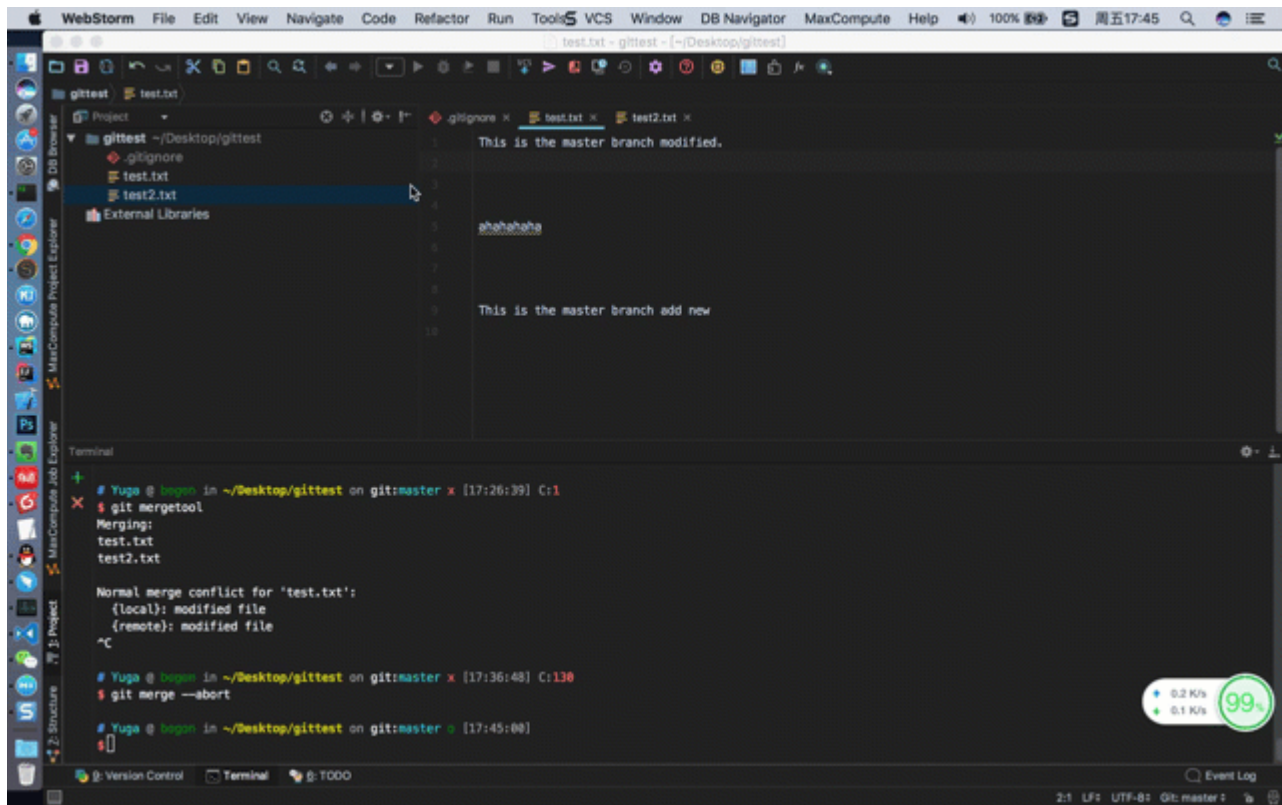
图中 **A** 为当前分支最新的一次提交，**B** 为当前本地代码，**C** 为需要合并的分支的代码。然后你可以通过点击相应的冲突点来修改合并后的输出结果，然后保存后，如果还有其他的冲突文件，会继续弹出此界面，进行相应的修改。

PS: 如果你不想每次 `git mergetool` 命令都需要输入你想使用的工具名称，可以设置工具为默认工具即可：

```
1 $ git config merge.tool kdiff3
```

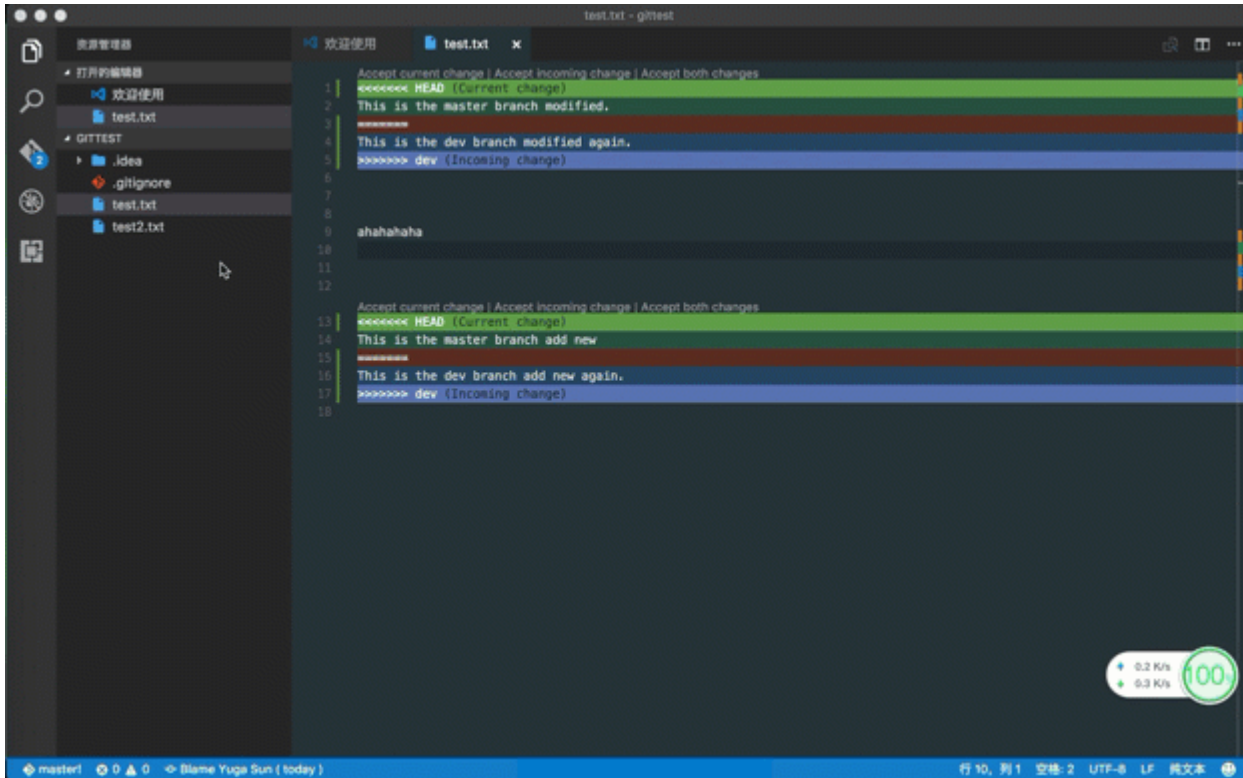
WebStorm 集成开发环境

webstorm是个很强大的集成开发工具，自然解决冲突的功能不在话下，直接上动画图，一看就明白了：



Visual Studio Code 开发工具

Visual Studio Code 也是个很强大的开发工具，解决冲突也很简单高效，前提是你需要安装 `Better Merge` 插件，直接上动画图，一看就明白了：



直接废弃 git merge 操作

如果你不想要本次merge 分支的代码，直接 `abort` 即可

```
1 $ git merge --abort
```

使用 abort 个人觉得始终逃避问题的方式，真正的勇士应该敢于直面惨淡的人生，遇到冲突，就去解决它，而不是逃避它。

后记

解决的方法很多，我这里只是列出了个人比较常用的几个方法，如果大家有更好的方法，可以随时分享。