

# Git冲突与解决方法

## 1、git冲突的场景

- 情景一：多个分支代码合并到一个分支时；
- 情景二：多个分支向同一个远端分支推送代码时；

实际上，push操作即是把本地代码merge到远端库分支上。

关于push和pull其实就分别是用本地分支合并到远程分支 和 将远程分支合并到本地分支

所以这两个过程中也可能存在冲突。

git的合并中产生冲突的具体情况：

<1>两个分支中修改了同一个文件（不管什么地方）

<2>两个分支中修改了同一个文件的名称

两个分支中分别修改了不同文件中的部分，不会产生冲突，可以直接将两部分合并。

## 2、冲突解决方法

- 情景一：在当前分支上，直接修改冲突代码--->add--->commit。
- 情景二：在本地当前分支上，修改冲突代码--->add--->commit--->push

注：借用vim或者IDE或者直接找到冲突文件，修改。

## 3、实战演示

### (1) 情景

本地库中两个不同分支，修改同一个文件同一代码块，两分支先后将修改合并到master分支上，master在合并第二个分支代码时，报错：合并冲突。

## (2) 本地库

### <1>master分支

```
Administrator@C:\Users\Administrator> cd /d/1gitTest
$ git init
Initialized empty Git repository in D:/1gitTest/.git/

Administrator@C:\Users\Administrator> cd /d/1gitTest (master)
$ touch mergeTest.txt

Administrator@C:\Users\Administrator> cd /d/1gitTest (master)
$ vim mergeTest.txt
```

```
Administrator@C:\Users\Administrator> cd /d/1gitTest (master)
$ cat mergeTest.txt
hello master branch!

Administrator@C:\Users\Administrator> cd /d/1gitTest (master)
$ git add .

Administrator@C:\Users\Administrator> cd /d/1gitTest (master)
$ git commit -m 'master commit'
[master (root-commit) 75d944e] master commit
1 file changed, 1 insertion(+)
create mode 100644 mergeTest.txt
```

### <2>建立两个分支

```
Administrator@C:\Users\Administrator> cd /d/1gitTest (master)
$ git checkout -b bBranch
Switched to a new branch 'bBranch'

Administrator@C:\Users\Administrator> cd /d/1gitTest (bBranch)
$ git checkout -b aBranch
Switched to a new branch 'aBranch'

Administrator@C:\Users\Administrator> cd /d/1gitTest (aBranch)
$ git branch -a
* aBranch
  bBranch
  master
```

### <3>两分支修改提交

aBranch分支：

```
Administrator@ubuntu:~/1gitTest (aBranch)
$ cat mergeTest.txt
hello master branch!

Administrator@ubuntu:~/1gitTest (aBranch)
$ vim mergeTest.txt
```

```
Administrator@ubuntu:~/1gitTest (aBranch)
$ cat mergeTest.txt
hello master-aBranch branch!
```

```
Administrator@ubuntu:~/1gitTest (aBranch)
$ git add mergeTest.txt
```

```
Administrator@ubuntu:~/1gitTest (aBranch)
$ git commit -m 'aBranch commit'
[aBranch 5b7fa94] aBranch commit
1 file changed, 1 insertion(+), 1 deletion(-)
```

bBranch分支：

```
Administrator@ubuntu:~/1gitTest (aBranch)
$ git checkout bBranch
Switched to branch 'bBranch'
```

```
Administrator@ubuntu:~/1gitTest (bBranch)
$ cat mergeTest.txt
hello master branch!
```

```
Administrator@ubuntu:~/1gitTest (bBranch)
$ vim mergeTest.txt
```

```
Administrator@ubuntu:~/1gitTest (bBranch)
$ cat mergeTest.txt
hello master-bBranch branch!
```

```
Administrator@ubuntu:~/1gitTest (bBranch)
$ git add .
```

```
Administrator@ubuntu:~/1gitTest (bBranch)
$ git commit -m 'bBranch commit'
[bBranch 7adfff1] bBranch commit
1 file changed, 1 insertion(+), 1 deletion(-)
```

### (3) 合并分支产生冲突

合并aBranch分支（将aBranch分支合并到当前master分支上）：

```
Administrator@ /d/1gitTest (bBranch)
$ git checkout master
Switched to branch 'master'

Administrator@ /d/1gitTest (master)
$ git status
On branch master
nothing to commit, working tree clean

Administrator@ /d/1gitTest (master)
$ cat mergeTest.txt
hello master branch!
```

```
Administrator@ /d/1gitTest (master)
$ git merge --no-ff aBranch
Merge made by the 'recursive' strategy.
 mergeTest.txt | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

Administrator@ /d/1gitTest (master)
$ git log --graph
*      commit c031260660ab9db6c7311395ec108b9cb00d28ca (HEAD -> master)
|\     Merge: 75d944e 5b7fa94
|      Author: Gavin <xsh003@126.com>
|      Date:   Tue May 22 11:24:28 2018 +0800
|      |
|      Merge branch 'aBranch'
|      |
|      *      commit 5b7fa947624b0b0fa4284293153537d904aad297 (aBranch)
|      /      Author: Gavin <xsh003@126.com>
|      /      Date:   Tue May 22 11:17:19 2018 +0800
|      |
|      aBranch commit
|      |
|      *      commit 75d944e904727673c287c1335620a48b7622853a
|      /      Author: Gavin <xsh003@126.com>
|      /      Date:   Tue May 22 11:12:32 2018 +0800
|      |
|      master commit
```

注：

git merge：默认情况下，Git执行"快进式合并"（fast-forward merge），会直接将Master分支指向Develop分支。  
使用--no-ff参数后，会执行正常合并，在Master分支上生成一个新节点。为了保证版本演进的清晰，建议采用这种方法。

再合并bBranch分支，产生冲突：

```
Administrator /d/1gitTest (master)
$ git merge --no-ff bBranch
Auto-merging mergeTest.txt
CONFLICT (content): Merge conflict in mergeTest.txt
Automatic merge failed; fix conflicts and then commit the result.
```

mergeTest.txt 文件内容：

```
<<<<<< HEAD
hello master-aBranch branch!
=====
hello master-bBranch branch!
>>>>>> bBranch
~
~
~
~
~
~
```

#### (4) 解决冲突

--->在当前分支上（master），找到冲突文件，直接修改冲突代码，add，commit。

```
Administrator /d/1gitTest (master | MERGING)
$ vim mergeTest.txt

Administrator /d/1gitTest (master | MERGING)
$ cat mergeTest.txt
hello master-aBranch-bBranch branch!

Administrator /d/1gitTest (master | MERGING)
$ git add .

Administrator /d/1gitTest (master | MERGING)
$ git commit -m 'fix conflict commit'
[master 1a0be1d] fix conflict commit
```

注：简单方法，使用vim修改，cat查看冲突文件。（注意要删除git自动生成的冲突代码分隔符）

#### (5) 完成冲突解决

```
Administrator /d/1gitTest (master)
$ git log --graph
* commit 1a0be1ddcf728ddb6336da56d8c012bb54babed1 (HEAD -> master)
  Merge: c031260 7adfff1
  Author: Gavin <xsh003@126.com>
  Date: Tue May 22 11:38:43 2018 +0800

    fix conflict commit

* commit 7adfff1b58e78ae0598e19165127d663d5d78cf9 (bBranch)
  Author: Gavin <xsh003@126.com>
  Date: Tue May 22 11:23:21 2018 +0800

    bBranch commit

* commit c031260660ab9db6c7311395ec108b9cb00d28ca
  Merge: 75d944e 5b7fa94
  Author: Gavin <xsh003@126.com>
  Date: Tue May 22 11:24:28 2018 +0800

    Merge branch 'aBranch'

* commit 5b7fa947624b0b0fa4284293153537d904aad297 (aBranch)
  Author: Gavin <xsh003@126.com>
  Date: Tue May 22 11:17:19 2018 +0800

    aBranch commit

* commit 75d944e904727673c287c1335620a48b7622853a
  Author: Gavin <xsh003@126.com>
  Date: Tue May 22 11:12:32 2018 +0800

    master commit
```

注：提交或者合并都会生成git节点。每个节点对应一个代码版本。