# Git如何在不提交当前分支的情况下切换到其它分支进行操作——git stash

****假如现在的Bug你还没有解决，而上边又给你派了一个新的Bug，而这个Bug相比较现在正在苦思冥想的Bug比较容易解决。****

****你想先解决新的Bug，可是之前的Bug还没有解决完而不能提交。怎么办?****

****解决方法****：在其他分支上另开炉灶解决。

首先你需要将此刻正在解决Bug的当前分支“储藏”起来。例假如此时正在你在当前分支dev上已进行了Bug修改但还未提交。

此时你想去解决刚派下来的另一个Bug。而你需要在master分支上去修复这个Bug，第一步就需要先切换到master 分支。当你执行****$ git checkout master****命令的时候，将提示出错：

****error: Your local changes to the following files would be overwritten by checkout:  
        readme.txt  
Please commit your changes or stash them before you switch branches.****

（请在切换分支之前提交您的更改或隐藏它们）

因为当前的分支dev 最初也是从master 分支上衍生出来的。而此时你要再从该分支上切换到其主分支。那么你需要先把该dev分支上的改动提交后才能切换，但是该dev分支上还没有完成全部的修改，你不想提交。那么此时你就要选择 stash 它们（你在当前分支上改动的却没有提交commit的内容）。

所以第二步，在当前分支上执行 ****$ git stash****命令。将当前分支存起来，id为 8528ea2 s

IMG_256

这时候再执行 $ git status 命令，显示没有东西需要提交，这个时候你就可以切换到master分支上了。

接着就可以在主分支master上创建并切换到新的分支去修复另一个Bug了。

那修改完那个Bug也提交后，就该回到dev 分支上去继续修改那个未完成的Bug。

执行  $ git checkout dev 切换到 dev 分支，这个时候执行 git status 命令仍旧显示没有东西需要提交。毕竟我们前边已经成功将dev上未提交的改动给“隐藏“了，这时，用****$ git stash list**** 命令****去查看我们“存储”的列表****。

IMG_257

会发现id 为  8528ea2 s 的储藏项目在列表中，我们需要将其恢复，有两个办法：

一、用****$ git stash apply**** 命令恢复，但是恢复后，stash内容并不删除，这时候再执行  $ git stash list 命令，id 为  8528ea2 s 的储藏项目还会在列表中，你需要用 $ git stash drop 来删除；

注意： 如果有一个分支上多个 stash，如果需要恢复指定的 stash ，可以在命令尾部加id，如  $ git stash apply ****stash@{0}****，同样删除指定 stash 项目则执行如 $ git stash drop****stash@{1}**** 。

二、用 ****$ git stash pop**** 命令，恢复的同时把 stash 存储列表的内容也删了。这时候再执行  $ git stash list 命令，id 为  8528ea2 s 的储藏项目不会在列表中。

此时再查看 $ cat <filename>  会发现之前的改动还存在，且执行 git status 就会继续显示该分支上有改动未提交。