

`git stash` 可用来暂存当前正在进行的工作，比如想pull 最新代码，又不想加新commit，或者另外一种情况，为了fix 一个紧急的bug，先stash，使返回到自己上一个commit，改完bug之后再stash pop，继续原来的工作。

基础命令：

**\$git stash**

**\$do some work**

**\$git stash pop**

在使用git pull代码时，经常会碰到有冲突的情况，提示如下信息：

```
error: Your local changes to 'c/envIRON.c' would be overwritten by merge.  Aborting.  
Please, commit your changes or stash them before you can merge.
```

这个意思是说更新下来的内容和本地修改的内容有冲突，先提交你的改变或者先将本地修改暂时存储起来。

处理的方式非常简单，主要是使用git stash命令进行处理，分成以下几个步骤进行处理。

## 1、先将本地修改存储起来

```
$ git stash
```

这样本地的所有修改就都被暂时存储起来。是用git stash list可以看到保存的信息：

```
stash@{0}: WIP on master: a80c24d 打印系统环境变量
```

git stash暂存修改

其中stash@{0}就是刚才保存的标记。

## 2、pull内容

暂存了本地修改之后，就可以pull了。

```
$ git pull
```

## 3、还原暂存的内容

```
$ git stash pop stash@{0}
```

系统提示如下类似的信息：

```
Auto-merging c/environ.c
CONFLICT (content): Merge conflict in c/environ.c
```

意思就是系统自动合并修改的内容，但是其中有冲突，需要解决其中的冲突。

#### 4、解决文件中冲突的部分

打开冲突的文件，会看到类似如下的内容：

```
<<<<<< Updated upstream
    int i, j;
=====
    int i, z;
>>>>>> Stashed changes
```

git冲突内容

其中Updated upstream 和=====之间的内容就是pull下来的内容，=====和stashed changes之间的内容就是本地修改的内容。碰到这种情况，git也不知道哪行内容是需要，所以要自行确定需要的内容。

解决完成之后，就可以正常的提交了。

进阶：

**Git stash save** "work in progress for foo feature"

当你多次使用'**git stash**'命令后，你的栈里将充满了未提交的代码，这时候你会对将哪个版本应用回来有些困惑，

'**git stash list**' 命令可以将当前的Git栈信息打印出来，你只需要将找到对应的版本号，例如使用'**git stash apply stash@{1}**'就可以将你指定版本号为**stash@{1}**的工作取出来，当你将所有的栈都应用回来的时候，可以使用'**git stash clear**'来将栈清空。

```
git stash          # save uncommitted changes
# pull, edit, etc.
git stash list     # list stashed changes in this git
git show stash@{0} # see the last stash
git stash pop      # apply last stash and remove it from the list

git stash --help   # for more info
```

