情景：

有两个用户us1和us2，他们都从服务器上克隆了同一个git工程下来，然后各自在本地电脑上对此工作进行修改。假设us1的远程版本库名字为user1，us2的远程版本库名字为user2，这里user1和user2指向的是同一个服务器的地址，假设为//10.15.14.97. 且他们要push的分支为master分支。远程库的默认版本库使用默认的名称origin，相当于主机名，其默认分支为master.

us1首先完成属于他自己的工作，于是他把自己的修改push到服务器上，由于在us1之前没有人对服务器上的工程进行过更新，所以服务器允许us1把自己的分支push上来。服务器上的工程更新后，us2也完成了自己的修改工作，他也想把自己的修改push到服务器上，于是他输入以下命令：

$ git push user2 master

然后系统提示：

To //10.15.14.97/test/project/backup.git/

! [rejected] master -> master (fetch first)

error: failed to push some refs to '//10.15.14.97/test/project/backup.git/'

hint: Updates were rejected because the remote contains work that you do

hint: not have locally. This is usually caused by another repository pushing

hint: to the same ref. You may want to first integrate the remote changes

hint: (e.g., 'git pull ...') before pushing again.

hint: See the 'Note about fast-forwards' in 'git push --help' for details.

这时因为现在服务器上的版本要比us2的master分支上的版本要新，只能通过融合后才能把自己的更新push上去。

解决冲突的办法是首先从远程分支中获取最新版本到本地上，通过融合后把冲突解决，最后再push，更新本地版本有如下两个命令：

1. **git fetch：从远程获取某个分支的最新版本到本地，但不会自动合并merge.**

回到情景中，us2用户可以先输入如下命令：

$ git fetch origin master

git fetch命令格式为：

git fetch <远程主机名> <分支名>

输出以下信息：

remote: Counting objects: 3, done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 1), reused 0 (delta 0)

Unpacking objects: 100% (3/3), done.

From //10.15.14.97/test/project/backup

\* branch master -> FETCH\_HEAD

b76a1da..56594fe master -> origin/master

这时已把本地的origin/maste远程分支(本地也是存有远程分支的，在克隆的时候已经存进)更新为服务器上的master分支。这时可以通过以下命令查看与origin/master的区别：

$ git diff origin master

然后进行合并，只有合并之后才能提交：

$ git merge origin master

执行完后，系统提示有冲突：

Auto-merging 4.txt

CONFLICT (content): Merge conflict in 4.txt

Automatic merge failed; fix conflicts and then commit the result.

提示冲突在4.txt文件上，这时你可以查看一下4.txt，会有以下文本：

<<<<<<< HEAD

888

=======

777

>>>>>>> origin

888是当前分支4.txt的内容，777是远程版本master中4.txt的内容，他们之间有冲突，于是us1和us2通过私下交流整合，决定把4.txt的内容修改为 888 777，修改之后由us2提交到本地上，然后再push到服务器上：

$ git add .

$ git commit -m "解决冲突2"

$ git push user2 master

Counting objects: 6, done.

Delta compression using up to 4 threads.

Compressing objects: 100% (4/4), done.

Writing objects: 100% (6/6), 531 bytes | 0 bytes/s, done.

Total 6 (delta 2), reused 0 (delta 0)

To //10.15.14.97/test/project/backup.git/

56594fe..6b5f1de master -> master

这时服务器已经更新为us2的最新版本了。Us1用户如果想更新为最新版本，一样可以先fetch,然后再merge。

1. **git pull：从远程获取最新版本并merge到本地。**

$ git pull origin master

上述命令其实相当于git fetch 和 git merge，在实际使用中，git fetch更安全一些因为在merge前，我们可以查看更新情况，然后再决定是否合并