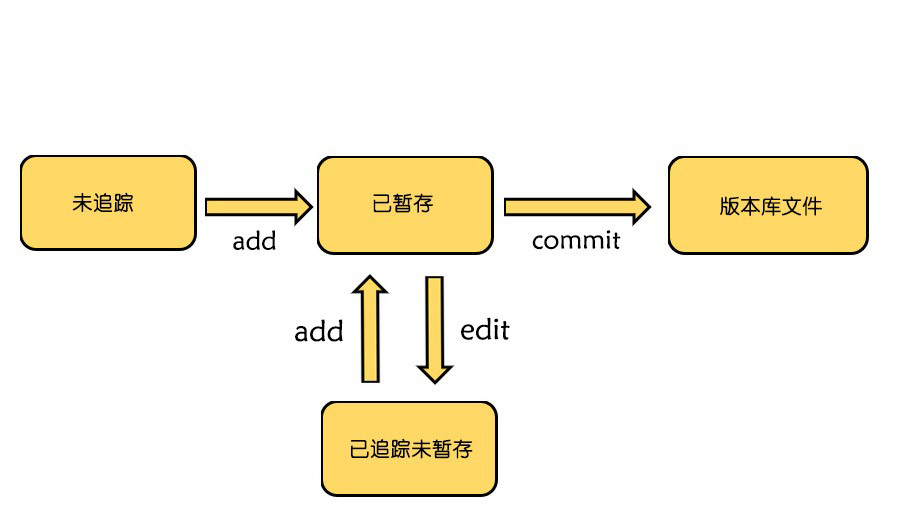
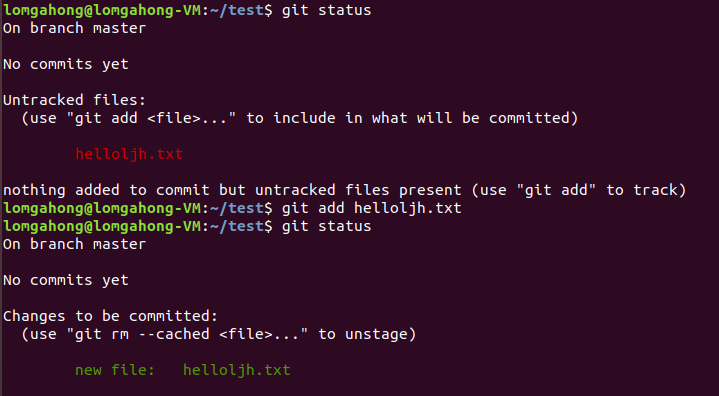
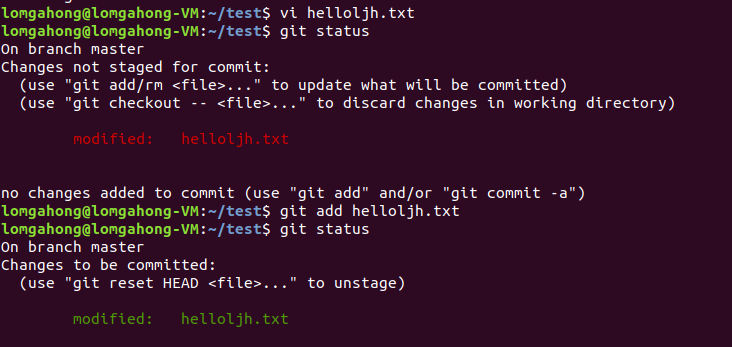
1. **git的工作流程**

* **文件的暂存与提交**

****

* 1. 当我们新建一个文件的时候，文件处于未被追踪状态。
  2. 用git add可以把文件添加到暂存区中，这时文件处于**已追踪已暂存**状态。
  3. 使用git commit可以把**已追踪已暂存**状态的文件推送到版本库中。
  4. 如果暂存区的文件被编辑，会变为**未暂存**状态，不可以被commit到版本库。
  5. 使用git commit可以把**已追踪未暂存**状态的文件转化为**已追踪已暂存**状态。





* **仓库的获取与推送**

如果要参与一个已有的团队项目，首先使用git clone来获取一个远程仓库的副本，在自己电脑中进行数据修改和更新后使用git pull拉取最新的版本库数据合并到自己电脑中的版本库，最后使用git push+仓库+分支，将自己电脑上的版本库上传到远程仓库。

1. **关于git的分支**

* **原理**

1. git中的分支，实际上是一个指向提交对象的指针，一个分支的指针会指向此分支中最新提交的对象。
2. 创建新分支的时候，其实是在当前位置新创建了一个可以移动的指针。
3. git通过一个叫HEAD的特殊指针，来判断当前操作的分支是哪一个分支，切换分支的操作其实是对HEAD指针的操作。
4. 分支之间的数据不会相互影响，随着修改次数的增加，同一点出发的两个分支的差异会越来越大。
5. 同一支路上的分支合并时，采用“快进模式”，也就是直接把往前合并，所以合并速度非常快。所以在Git上可以使用分支完成某个任务，合并后再删掉分支，这和直接在master分支上工作效果是一样的，但过程更安全
6. 不同支路上的分支合并时，git会使用两个分支末端所指的快照以及两个分支的工作祖先，进行一个简单的三方合并。git可以自行选择最优的共同工作祖先。

* **如何恰当地合并git分支**

1. 合并冲突的预防和解决

在工作中应该尽量避免在两个分支中对同一个文件的同一个部分进行不同的修改，否则git无法判断哪一个版本是用户想保留的版本，合并会暂停。此时可以使用 git status 命令来查看那些因包含合并冲突而处于未合并状态的文件，对文件进行修改后通过手动git add来解决冲突。

1. 分支合并后可删除已不再使用的分支文件