<https://www.cnblogs.com/sgdd123/p/8338403.html>

之前在windows中一直采用github的桌面版，库的建立更新都是借助软件的帮助。所使用的的功能也非常局限，仅仅只是创建库再提交自己的代码。至于版本管理、回滚、分支以及git的结构都没有清楚的认识。这次由于任务需要，在ubuntu中必须使用git来链接github，所以必须接触一些git的命令，也对git的思想有了进一步的认识。这里简单梳理以下，作为备忘录。文末也附上我参考的博客，有兴趣的可以进一步阅读。

## 一些认识

之前对github和git一直傻傻分不清，以为是两种不同的版本管理工具。现在才认识到git是一种版本管理工具，而github只是一个远程仓库。实际上没有github这种远程仓库，你计算机中git也会建立一个本地的仓库，或者链接到其他的远程仓库。

## 1. git的安装

输入命令安装git，并查看版本以确认安装成功。

sudo apt-get install git

git --version

## 2. git的使用

先解释一些概念，git分四层管理代码。

1. 你目录中的文件是第一层
2. 缓存区，每次add之后，当前目录中要追踪的文件会作为一个版本会存放在缓存区。注意不是所有的文件。一般一个文件生成之后，会标记为“未追踪”，但是否对其做版本管理还是要选择的。例如一些编译文件就没有必要追踪。对需要做版本管理的问件，用add添加，不需要的用clean删除。
3. 本地仓库，每次commit之后，缓存区最新的版本就会存放在本地仓库。这里要提及一个HEAD的概念。HEAD是当前的版本指向，每次更新或者回退都会修改HEAD的指向，但对仓库中每一个版本并不会删除。所以即使回退到过去还是有机会回到现在的版本的。
4. 远程仓库，每次push之后，会将本地仓库中HEAD所指向的版本存放到远程仓库

这里附上一些我常用的命令作为备忘录，详细的使用效果请参考我文末附上的博客链接。

| **命令** | **功能** |
| --- | --- |
| git init | 在本地的当前目录里初始化git仓库 |
| git status | 查看当前仓库的状态 |
| git add -A | 增加目录中所有的文件到缓存区 |
| git add file | 增加相应文件到缓存区 |
| git commit -m "信息" | 将缓存区中更改提交到本地仓库 |
| git log | 查看当前版本之前的提交记录 |
| git reflog | 查看HEAD的变更记录，包括回退 |
| git branch -b branch\_name | 建立一个新的分支 |
| git diff | 查看当前文件与缓存区文件的差异 |
| git checkout -- file | 取消更改，将缓存区的文件提取覆盖当前文件 |
| git reset --hard 版本号 | 回退到相应版本号，同样也可以回退到未来的版本号 |
| git clean -xf | 删除当前目录中所有未追踪的文件 |
| git config --global core.quotepath false | 处理中文文件名 |

这些命令用于处理本地仓库的版本管理足够了，下面我介绍与远程仓库打交道。

## 2. git与github的链接

首先我们认为你已经有一个github的账户。

然后我们要建立SSH链接。这是一种通讯的加密协议。我先在我的笔记本上计算一对公钥和私钥，将公钥存储在github中，这样本地就可以通过SSH与github展开加密通讯。详细的内容可以参考[SSH原理与运用（一）：远程登录](http://www.ruanyifeng.com/blog/2011/12/ssh_remote_login.html)。

建立方法，输入命令

ssh-keygen -t rsa -C "your\_email@youremail.com" //双引号里面是你的常用邮箱

输入之后要输入口令，可以不用输入直接按“enter”一路确认就可以了。然后在账户的根目录（/或者/home/你的账户名，具体取决于你执行上述命令时所采用的账户）查找隐藏目录.ssh/id\_rsa.pub文件，将当中内容添加到github中。

这样你就可以通过SSH链接到github中了。但是github作为一个远程仓库，你可以链接这个仓库，并保持同步。但是你不能把本地仓库直接上传到github中去。所以你应该先在github中建立一个对应的仓库，然后再在本地建立一个仓库，将两者进行链接，再去写入文件执行版本管理。所用到的命令有

git remote add origin git@github.com:<用户名>/<仓库名>.git

git pull origin master //因为github建立仓库时会有readme.md文件，先要拷贝一份

git push -u origin master //将本地仓库链接到master分支上，你当然可以链接到其他分支

git push//上传你的本地仓库

还有一种方法不用分两地建库再去链接。你可以只在github上建库，然后clone到本地目录中。

git clone git@github.com:<用户名>/<仓库名>.git

至于团队合作中的分支管理，由于现在还用不到，等以后有机会试用在去学习吧。