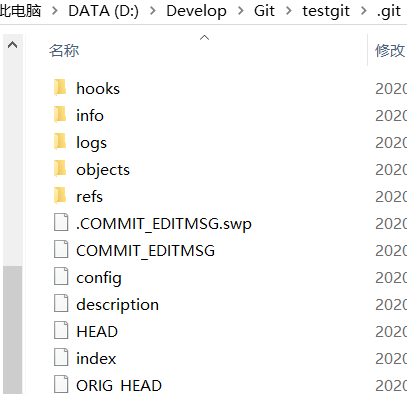
# 1 Git

Git是目前世界上最先进的分布式版本控制系统。

## 1.1 工作原理/流程





Workspace：工作区  
Index / Stage：暂存区  
Repository：仓库区（或本地仓库）  
Remote：远程仓库

**工作区**：就是你在电脑上看到的目录，比如目录下testgit里的文件(.git隐藏目录版本库除外)。或者以后需要再新建的目录文件等等都属于工作区范畴。

**版本库(Repository)**：工作区有一个隐藏目录.git,这个不属于工作区，这是版本库。其中版本库里面存了很多东西，其中最重要的就是**stage(暂存区)**，还有Git为我们自动创建了第一个分支master,以及指向master的一个指针HEAD。

Git提交文件到版本库有两步：

第一步：是使用 git add 把文件添加进去，实际上就是把文件添加到暂存区。

第二步：使用git commit提交更改，实际上就是把暂存区的所有内容提交到当前分支上。

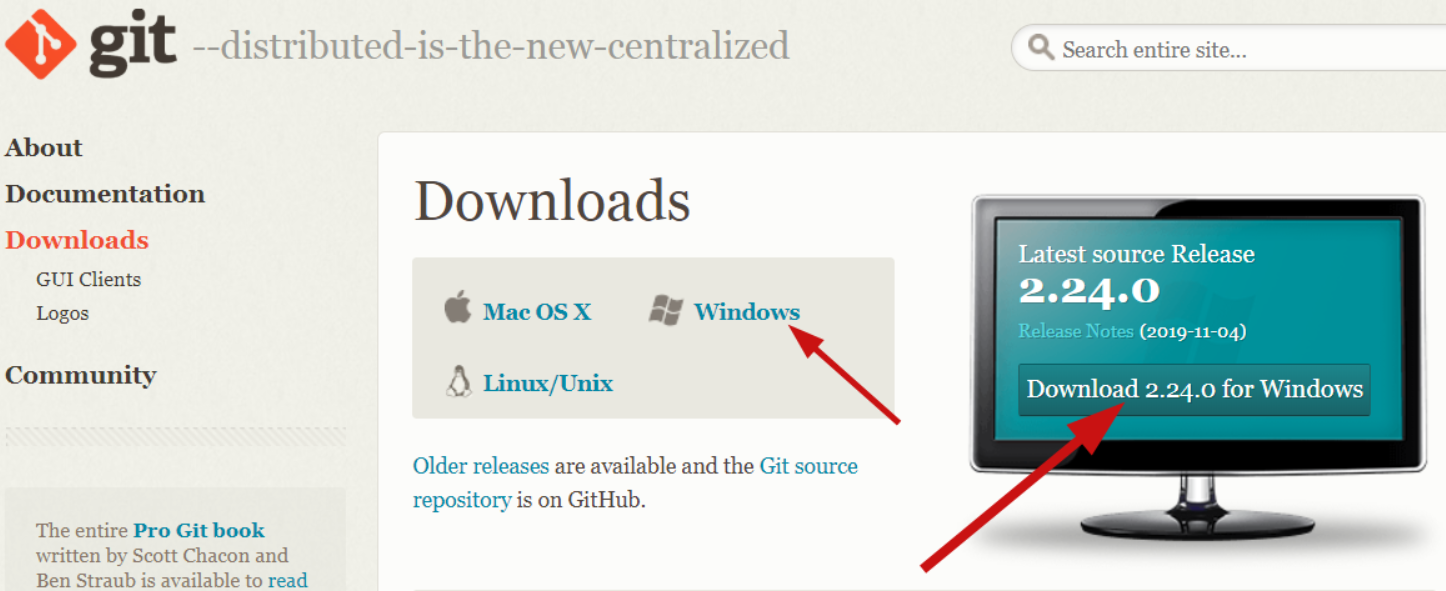
## 1.2 SVN与Git的最主要的区别

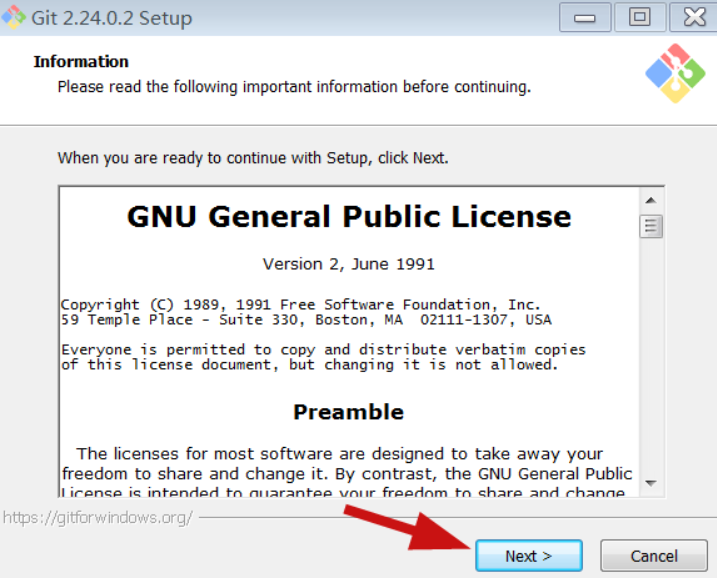
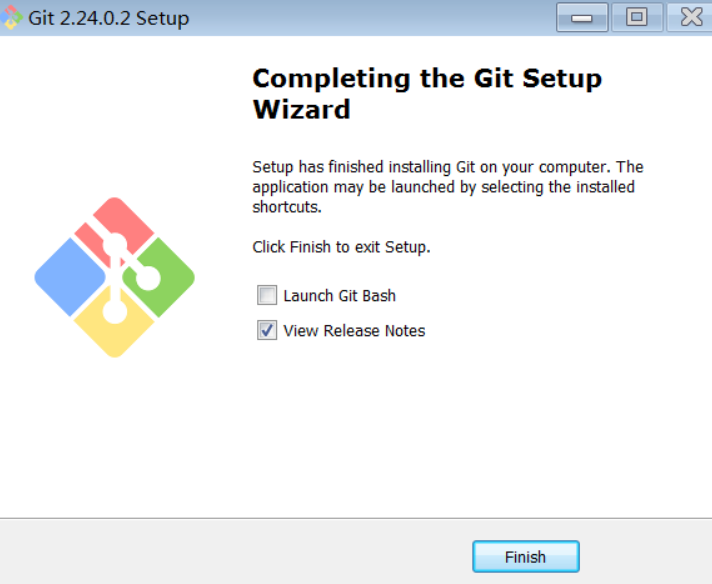
SVN是集中式版本控制系统，版本库是集中放在中央服务器的，而干活的时候，用的都是自己的电脑，所以首先要从中央服务器哪里得到最新的版本，然后干活，干完后，需要把自己做完的活推送到中央服务器。集中式版本控制系统是必须联网才能工作，如果在局域网还可以，带宽够大，速度够快，如果在互联网下，如果网速慢的话，就纳闷了。

Git是分布式版本控制系统，那么它就没有中央服务器的，每个人的电脑就是一个完整的版本库，这样，工作的时候就不需要联网了，因为版本都是在自己的电脑上。既然每个人的电脑都有一个完整的版本库，那多个人如何协作呢？比如说自己在电脑上改了文件A，其他人也在电脑上改了文件A，这时，你们两之间只需把各自的修改推送给对方，就可以互相看到对方的修改了。

## 1.3 在windows上如何安装Git

1）到Git官网下载，网站地址：https://git-scm.com/downloads



2）启动测试

Git的安装完成，可以在开始菜单中看到Git的三个启动图标：

Git BashGit——配套的一个控制台；

Git CMD(Deprecated)——通过CMD使用Git（不推荐使用）；

Git GUI）——Git的可视化操作工具。

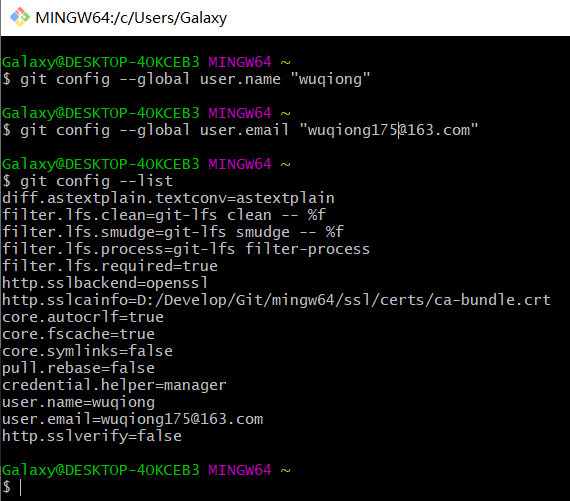
3）安装完成后，还需要最后一步设置

因为Git是分布式版本控制系统，所以需要填写用户名和邮箱作为一个标识。

注意：git config --global 参数，有了这个参数，表示你这台机器上所有的Git仓库都会使用这个配置，当然你也可以对某个仓库指定的不同的用户名和邮箱。

$ git config --global user.name "wuqiong"

$ git config --global user.email "wuqiong175@163.com"

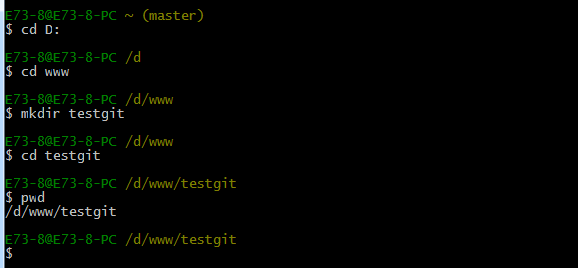


## 1.4 操作Git（本地）

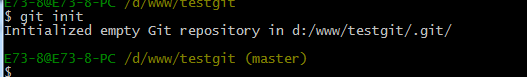
1）创建版本库

版本库又名仓库，英文名repository,你可以简单的理解一个目录，这个目录里面的所有文件都可以被Git管理起来，每个文件的修改，删除，Git都能跟踪，以便任何时刻都可以追踪历史，或者在将来某个时刻还可以将文件“还原”。

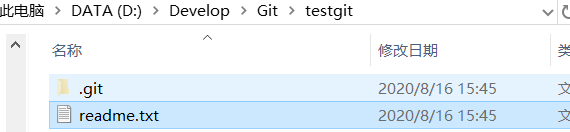
新建一个testgit版本库：



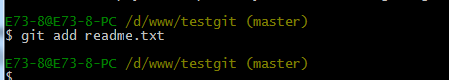
通过命令 git init 把这个目录变成git可以管理的仓库（.git的目录）：



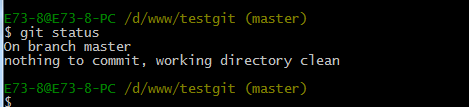
新建一个记事本文件 readme.txt 内容如下：11111111：



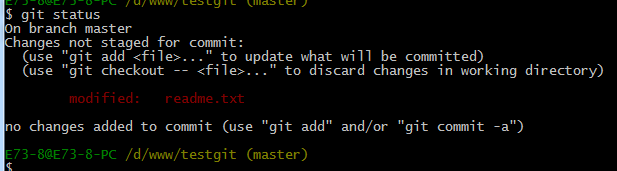
用命令 git commit告诉Git，把文件提交到仓库：



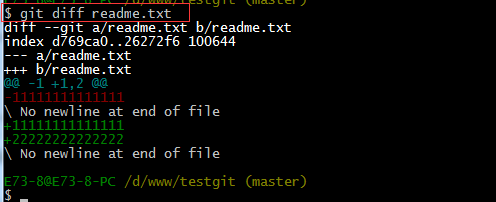
可以通过命令git status来查看是否还有文件未提交



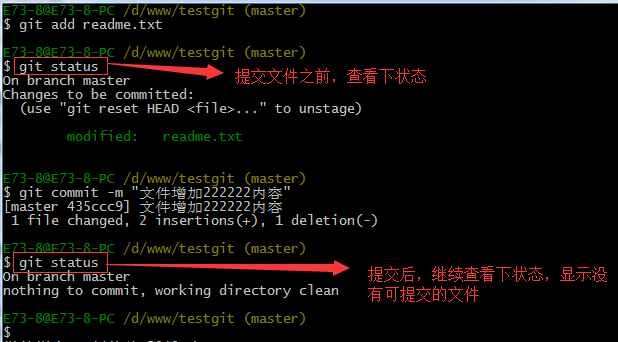
改下readme.txt内容，比如我在下面添加一行2222222222内容，继续使用git status来查看下结果（readme.txt文件已被修改，但是未被提交的修改）：



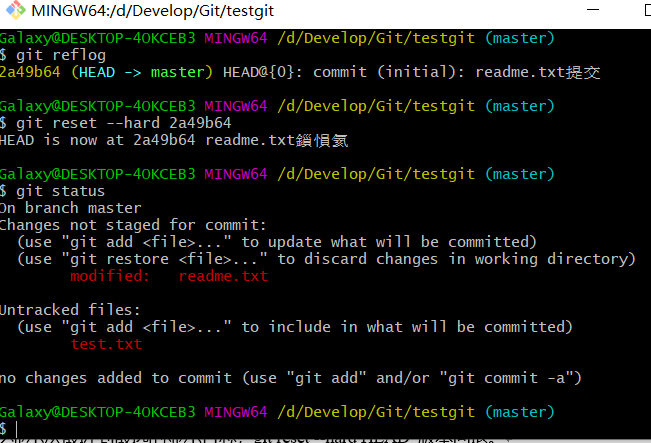
git diff readme.txt：

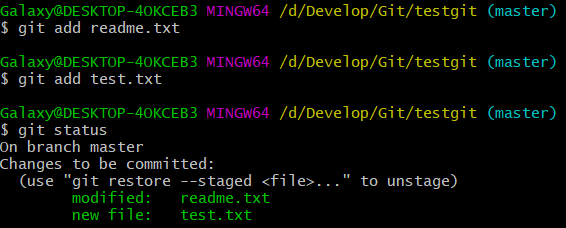


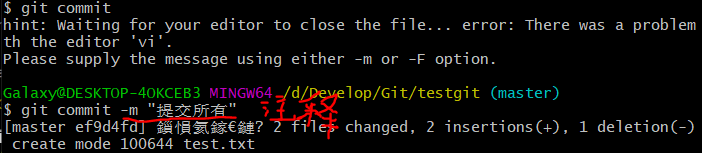
提交修改和提交文件是一样的2步(第一步是git add 第二步是：git commit)：

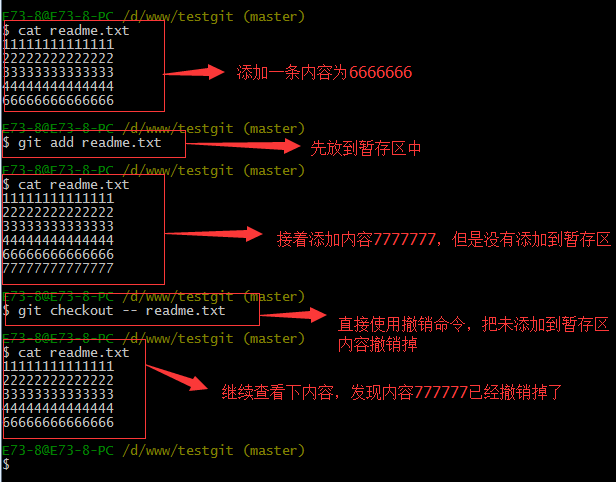


git log命令显示从最近到最远的显示日志；cat 查看；git reset --hard HEAD^版本回退；git checkout -- file 可以丢弃工作区的修改。

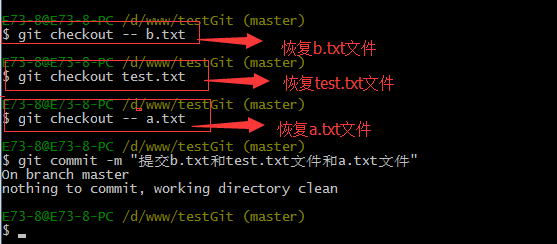








命令git checkout -- readme.txt 中的 -- 很重要，如果没有 -- 的话，那么命令变成创建分支了。只要没有commit之前，如果我想在版本库中恢复此文件。



## 1.5 远程仓库

### 1.5.1 配置git

先注册github账号，由于你的本地Git仓库和github仓库之间的传输是通过SSH加密的，所以需要一点设置：

1）设置用户名称和登录邮箱：

$ git config --global user.name "wuqiong"

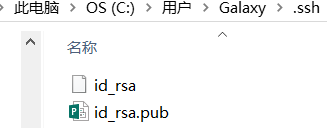
$ git config --global user.email "wuqiong175@163.com"

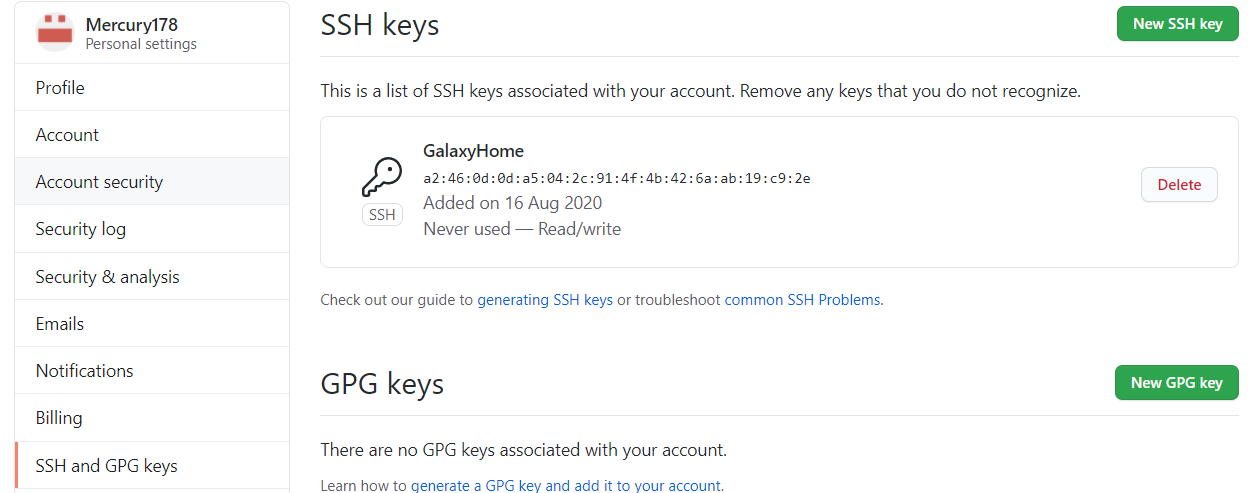
2）生成密钥：

ssh-keygen -t rsa -C "wuqiong175@163.com"

3）配置：

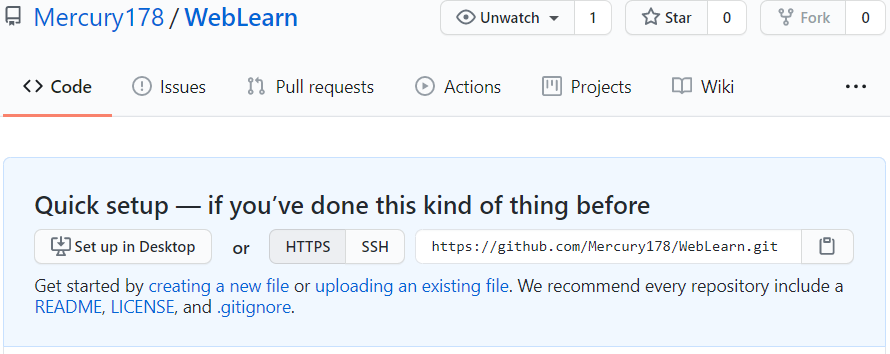
打开.ssh文件夹下id\_rsa.pub文件，复制内容。然后，登录Git服务器GitHub个人账户，在SSH keys菜单项中粘贴即可。





### 1.5.2 创建新项目/托管本地项目/克隆已有项目

1）创建远端仓库：



2）初始化本地项目&&添加项目

在Git客户端或CMD命令行，进入本地项目目录下，执行如下命令：

git init生成了.git隐藏文件。

