

git简易攻略

一、版本控制系统

版本控制是指对软件开发过程中各种程序代码、配置文件及说明文档等文件变更的管理，是软件配置管理的核心思想之一。版本控制最主要的功能就是追踪文件的变更。它将什么时候、什么人更改了文件的什么内容等信息忠实地记录下来。每一次文件的改变，文件的版本号都将增加。除了记录版本变更外，版本控制的另一个重要功能是并行开发。软件开发往往是多人协同作业，版本控制可以有效地解决版本的同步以及不同开发者之间的开发通信问题，提高协同开发的效率。并行开发中最常见的不同版本软件的错误(Bug)修正问题也可以通过版本控制中分支与合并的方法有效地解决。

目前的版本控制系统分为集中式和分布式的。集中式版本控制系统，在企业中用的比较多的是SVN，SVN是Subversion的简称，它是一个开源的版本控制系统。分布式版本控制系统最好用的就是现在风靡全球的Git。我们接下来学习就是它了。

SVN与Git最主要的区别有哪些呢？

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，你们两人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。

二、Git的安装

2.1 Linux

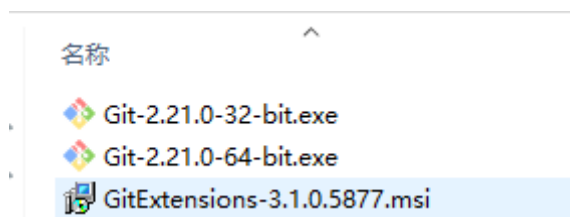
```
$sudo apt-get install git
```

2.2 Windows

下载Git, GitExtensions

<https://git-scm.com/download/win>

<https://github.com/gitextensions/gitextensions>



在Windows上安装时，先安装Git，再安装GitExtensions。一路默认安装即可

三、Git的使用

3.1 初始化git仓库

```
$ cd project1
$ git init
```

3.2 配置用户名和邮箱

```
$ git config --global user.name "Your Name"
$ git config --global user.email "email@example.com"
```

3.2 往仓库中添加文件

```
$ touch readme.txt
$ vim "编辑readme.txt"                                "在工作区中进行"

$ git add readme.txt                                    "在工作区的操作添加到暂存区"
$ git commit -m "lwh: add file readme.txt"              "将暂存区中的内容提交到版本库"

"对于有很多一次性同时添加"
$ git add .
$ git commit -m "..."
```

3.3 查看库的状态

```
$ git status
$ git log "查看库的版本号"
$ git reset --hard c5674dd41
```

3.4 推送到远程仓库

```
$ cd ~
$ ssh-keygen -t rsa -C "youremail@example.com"
$ cd ~/.ssh/
$ cat id_rsa.pub "得到如下的内容，之后，需要把这一长串字符串添加到github上配置SSH的位置"
ssh-rsa
AAAAB3NzaC1yc2EAAAADAQABAAQAC7refxr19nwsJms8ySaurGpX1FpWcyoiD0RG4e6oQisvE0nbQy
y7/CMYYI0niirGXzppBxte208/cjpkvqw4kIZBk9EBFtgmTsst57+/DV3mmsGpoXMNwIRd1n1ZrWtyT4
MDjqs7vEQgk2ndyU6ny8Dwm69IGEC8s5Zb8Ke02/2MSI0iwhPqsYfqYDpvWvo4MKIVvcsXuiwvNxCk1c
5JzzHr005eLijt40ej3FyqFSQsv5w2pf+fRWv9nGw4YdaSZxm0j2ycJlL8hDanGPhlUrD+xx3ZPyRHud
r4XiCDs5muesENkGcYitUKx0a0Rknogh93xGlnI/Z+i1fwCCMXKN haohb13@gmail.com
```

```
$ cd project1
$ git remote add origin git@github.com:haohb13/test.git # 给远程库起一个别名origin
$ git push -u origin master #将本地库推送到远程库

$ git remote remove origin # 删除origin库
$ git remote add origin git@gitee.com:haohb2020/test.git # 再重新给远程库起别名
```

3.5 版本回退

```
$ git log --pretty=oneline
a84cdf5bcb3d9ab7c219640376fe7216a9be1497 (HEAD -> master, origin/master) lwh:add
mvc
ae633a211b207b4149a5bb682a7b78e317e0f1c7 lwh:add textquery
2be0e38baee066393feb8ed673438f98c316ce46 lwh:add shared_ptr/weak_ptr
f39f1e38efbdbfa356f0544d1acd274ac44f9fef lwh:add unique_ptr
cadffb8f9e6075995d911c667e5381705545d5ff lwh:add severval fiels

$ git reset --hard ae633a211b207b41
HEAD is now at ae633a2 lwh:add textquery
```

3.6 删除文件

```
$ git rm readme.txt
$ git commit -m "lwh:delete readme.txt"
```

四、Git免密使用 (https)

3.7.1 新建文件，并保存密码

```
$ touch ~/.git-credentials
$ vim ~/.git-credentials

# 在.git-credentials文件中添加内容
https://{username}:{password}@github.com
```

3.7.2 添加配置

```
# 在执行命令
$ cd ~
$ git config --global credential.helper store
```

3.7.3 查看配置

```
# 查看~/.git-credentials 文件下是否有了以下配置项
[credential]
    helper = store
```

五、git的实现原理

Git 仓库目录（历史版本库）是 Git 用来保存项目的元数据和对象数据库的地方。这是 Git 中最重要的部分，从其它计算机克隆仓库时，拷贝的就是这里的数据。

工作目录是对项目的某个版本独立提取出来的内容。这些从 Git 仓库的压缩数据库中提取出来的文件，放在磁盘上供你使用或修改。

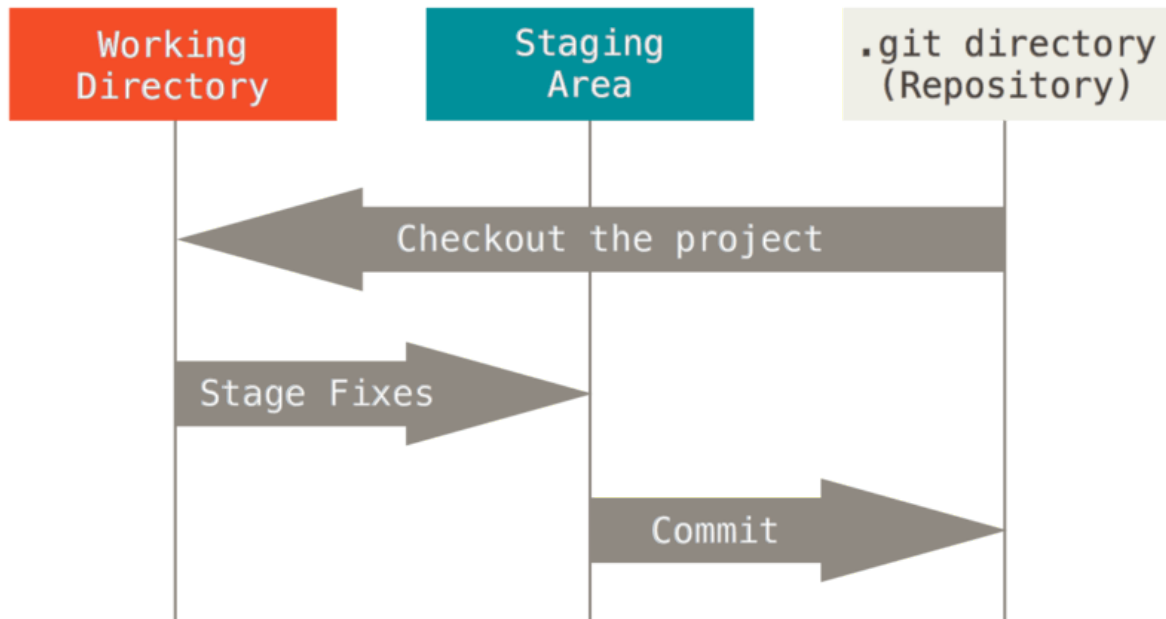
暂存区域是一个文件，保存了下次将提交的文件列表信息，一般在 Git 仓库目录中。有时候也被称作`索引`，不过一般说法还是叫暂存区域。

基本的 Git 工作流程如下：

在工作目录中修改文件。

暂存文件，将文件的快照放入暂存区域。

提交更新，找到暂存区域的文件，将快照永久性存储到 Git 仓库目录。



<https://marklodato.github.io/visual-git-guide/index-zh-cn.html>

<https://git-scm.com/book/en/v2>

六、git分支管理

6.1 创建分支

```
$ git checkout -b lwh
$ echo "this is lwh branch">lwh.txt
$ git add lwh.txt
$ git commit -m "lwh.txt"
```

6.2 切换分支

```
$ git checkout master
```

6.3 合并分支

```
$ git merge lwh    # 在当前分支合并lwh分支
```

6.4 删除分支

```
$ git branch -d lwh
```

6.5 查看分支历史

```
git log --graph --pretty=oneline --abbrev-commit
```

