

一、Git简介

Git是分布式版本控制系统

二、Git工作流程

- 1.从远程仓库中克隆Git资源作为本地仓库
- 2.从远程仓库中checkout代码然后进行代码修改
- 3.在提交前先将代码提交到暂存区
- 4.提交修改。提交到本地仓库。本地仓库中保存修改的各个历史版本
- 5.在修改完成后，需要团队成员共享代码时，可以将代码push到远程仓库

三、Git安装(windows)

最早Git是在Linux上开发的，很长一段时间内，Git也只能在Linux和Unix系统上跑。不过，慢慢地有人把它移植到了Windows上。现在，Git可以在Linux、Unix、Mac和Windows这几大平台上正常运行了

下载地址：<https://git-scm.com/download>

H:\学习资料\框架git\06-git\01.参考资料\环境安装包\64位windows

1.下载

2.安装git for windows

一路下一步，使用默认选项即可

3.安装TortoiseGit

- 1.一路下一步，使用默认选项即可；
- 2.默认选项会启动配置画面，只有英文语言包，默认即可继续下一步
- 3.配置git.exe，上面已经暗转管过git for window，所以在此找到git.exe所在目录
- 4.配置开发者姓名及邮箱，每次提交代码时都会把此信息包含到提交的信息中
- 5.使用默认配置，点击“完成”按钮完成配置。完整完毕后在系统右键菜单中会出现git的菜单项

4.安装中文语言包

文件名：TortoiseGit-LanguagePack-2.4.0.0-64bit-zh_CN.msi

H:\学习资料\框架git\06-git\01.参考资料\环境安装包\64位windows下的TortoiseGit-LanguagePack-2.4.0.0-64bit-zh_CN.msi

1.直接“下一步”完整完毕。

2.语言包安装完毕后可以在TortoiseGit的设置中调整语言

四、使用Git管理文件版本

1.创建版本库

1.版本库介绍

1.什么是版本库呢？版本库又名仓库，英文名repository，就是一个目录

2.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，以便任何时候都可以追踪历史，或者在将来某个时刻可以“还原”

3.由于git是分布式版本管理工具，所以git在不需要联网的情况下也具有完整的版本管理能力。

4.版本库：“.git**”目录就是版本库，将来文件都需要保存到版本库中。 **

工作目录：包含“.git**”目录的目录，也就是.git目录的上一级目录就是工作目录。只有工作目录中的文件才能保存到版本库中。

2.版本库创建步骤

1.创建一个版本库非常简单，可以使用git bash也可以使用tortoiseGit。首先，选择一个合适的地方，创建一个空目录

2.使用GitBash创建

在当前目录中点击右键中选择Git Bash来启动。

创建仓库执行命令：

```
$ git init
```

3.也可以使用TortoiseGit创建

使用TortoiseGit时只需要在目录中点击右键菜单选择“在这里创建版本库”，一般不勾选“制作纯版本库(没有工作目录)”

4.版本库创建成功，会在此目录下创建一个.git的隐藏目录

2.添加文件

1.在版本库所在的目录中创建一个mytest.txt文件，并写一些内容

2.右击mytest.txt文件，选择TortoiseGit->添加，之后该文本文件变为带"+ "号的图标

3.提交文件

右击带"+"号的文件，选择Git提交，日志内容必须填写：新增mytest.txt文件，单击提交

工作区和暂存区

工作区就是你在电脑里能看到的目录，比如我的reporstory文件夹就是一个工作区

把文件往Git版本库里添加的时候，是分两步执行的：

第一步是用git add把文件添加进去，实际上就是把文件修改添加到暂存区；

第二步是用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。

4.修改文件

1.提交修改

被版本库管理的文件不可避免的要发生修改，此时只需要直接对文件修改即可。修改完毕后需要将文件的修改提交到版本库。

在mytest.txt文件上点击右键，然后选择“提交”，填写日志内容：向mytest.txt中添加内容，单击提交

2.查看修改历史

在开发过程中可能会经常查看代码的修改历史，或者叫做修改日志。来查看某个版本是谁修改的，什么时间修改的，修改了哪些内容。

可以在文件上点击右键选择TortoiseGit->“显示日志”来查看文件的修改历史

3.差异比较

当文件内容修改后，需要和修改之前对比一下修改了哪些内容此时可以使用“比较差异功能”

可以在文件上点击右键选择TortoiseGit->“比较差异”来查看文件的修改前后的差异

4.还原修改

当文件修改后不想把修改的内容提交，还想还原到未修改之前的状态。此时可以使用“还原”功能

可以在文件上点击右键选择TortoiseGit->“还原”

注意：此操作会撤销所有未提交的修改，所以当做还原操作是需要慎重慎重！！

5.删除文件

需要删除无用的文件时可以使用git提供的删除功能直接将文件从版本库中删除。

可以在文件上点击右键选择TortoiseGit->“删除”

五、远程仓库

1.添加到远程库

在本地创建了一个Git仓库，又想让其他人来协作开发，此时就可以把本地仓库同步到远程仓库，同时还增加了本地仓库的一个备份

常用的远程仓库就是github

<https://github.com/>

1.在github上创建仓库

右上角"+"->New repository，填写Repository name: mytest。点击“create repository”按钮仓库就创建成功了

2.同步到远程仓库

同步到远程仓库可以使用git bash也可以使用tortoiseGit

1.使用git bash

1.在本地仓库所在的目录点击右键选择“Git Bash Here”，启动git bash程序。

2.然后在git bash中执行如下语句：

```
git remote add origin git@github.com:github个人用户名/mytest.git
```

```
git remote add origin https://github.com/VinsonFu/mytest1.git
```

```
git push -u origin master
```

3.若报错：remote origin already exists

可以先执行下面的命令：\$ git remote rm origin，再执行上面的命令：第2步

2.使用TortoiseGit

1.在windows下我们可以使用 Git Bash.exe来生成密钥，可以通过开始菜单或者右键菜单打开Git Bash

2.git bash 执行命令,生成公钥和私钥

命令: **ssh-keygen -t rsa**

3.执行命令完成后,在window本地用户.ssh目录C:\Users\用户名.ssh下面生成如下名称的公钥和私钥

4.密钥生成后需要在github上配置密钥本地才可以顺利访问。

在key部分将id_rsa.pub文件内容添加进去，然后点击“Add SSH key”按钮完成配置。

2.从远程仓库克隆

克隆远程仓库也就是从远程把仓库复制一份到本地，克隆后会创建一个新的本地仓库。选择一个任意部署仓库的目录，然后克隆远程仓库

1.使用git bash

```
$ git clone git@github.com:个人github用户名/mytest.git
```

2.使用TortoiseGit

在任意目录右击，选择Git考虑，填写url及目录

url: [git@github.com](https://github.com):个人github用户名/mytest.git

目录: 将要存放的目录

3.从远程仓库取代码

Git中从远程的分支获取最新的版本到本地有这样2个命令:

1. git fetch: 相当于是从远程获取最新版本到本地，不会自动merge (合并代码)

2.git pull: 相当于是从远程获取最新版本并merge到本地

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

如果使用TortoiseGit的话可以从右键菜单中点击“拉取” (pull) 或者“获取” (fetch)

4.搭建私有Git服务器

1.服务器搭建

搭建Git服务器需要准备一台运行Linux的机器，以CentOS为例

步骤:

1.安装git服务环境准备

```
yum -y install curl curl-devel zlib-devel openssl-devel perl cpio expat-devel gettext-devel gcc cc
```

2.下载git-2.5.0.tar.gz

H:\学习资料\框架git\06-git\01.参考资料\环境安装包\git服务器

a.解压缩

b.cd git -2.5.0

c. autoconf

d. ./configure

e.make

f.make install

3.添加用户

```
adduser -r -c 'git version control' -d /home/git -m git
```

此命令执行后会创建/home/git目录作为git用户的主目录

4.设置密码

password git

输入两次密码

5.切换到git用户

su git

6.创建git仓库

```
git --bare init /home/git/first
```

2.连接服务器

私有git服务器搭建完成后就可以向连接github一样连接使用了，但是我们的git服务器并没有配置密钥登录，所以每次连接时需要输入密码

使用命令连接：

```
$ git remote add origin ssh://git@192.168.25.156/home/git/first
```

这种形式和刚才使用的形式好像不一样，前面有ssh://前缀，也可以这样写：

```
$ git remote add origin git@192.168.25.156:first
```

六、分支管理

1.创建合并分支

1.每次的提交，Git都把它们串成一条时间线，这条时间线就是一个分支

2.在Git里，这个分支叫主分支，即master分支。HEAD指针严格来说不是指向提交，而是指向master，master才是指向提交的，所以，HEAD指向的就是当前分支

3.一开始的时候，master分支是一条线，Git用master指向最新的提交，再用HEAD指向master，就能确定当前分支，以及当前分支的提交点

4.每次提交，master分支都会向前移动一步，这样，随着你不断提交，master分支的线也越来越长。

5.当我们创建新的分支，例如dev时，Git新建了一个指针叫dev，指向master相同的提交，再把HEAD指向dev，就表示当前分支在dev上

6.Git创建一个分支很快，因为除了增加一个dev指针，改改HEAD的指向，工作区的文件都没有任何变化！

7.不过，从现在开始，对工作区的修改和提交就是针对dev分支了，比如新提交一次后，dev指针往前移动一步，而master指针不变

8.假如我们在dev上的工作完成了，就可以把dev合并到master上。Git怎么合并呢？最简单的方法，就是直接把master指向dev的当前提交，就完成了合并

2.使用Tortoise实现分支管理

1.创建分支

- 1.在本地仓库文件夹中点击右键，然后从菜单中选择“创建分支”
- 2.填写名称-分支：dev；基于-分支为当前分支
- 3.如果想创建完毕后直接切换到新分支可以勾选“切换到新分支”选项或者从菜单中选择“切换/检出”来切换分支

2.合并分支

- 1.分支切换到dev后就可以对工作区的文件进行修改，然后提交到dev分支原来的master分支不受影响。例如修改mytest.txt中的内容，然后提交到dev分支
- 2.切换到master分支后还是原来的内容
- 3.将dev分支的内容合并到master分支，当前分支为master。从右键菜单中选择“合并”
- 4.再查看mytest.txt的内容就已经更新了

3.解决冲突

- 1.两个分支中编辑的内容都是相互独立互不干扰的，那么如果在两个分支中都对同一个文件进行编辑，然后再合并，就有可能出现冲突。
- 2.例如在master分支中对mytest.txt进行编辑，然后提交到版本库；切换到dev分支，对mytest.txt进行编辑
- 3.最后进行分支合并，例如将dev分支合并到master分支。需要先切换到master分支然后进行分支合并，出现版本冲突，冲突需要手动解决
- 4.在冲突文件上单机右键选择“解决冲突”菜单项，把冲突解决完毕的文件提交到版本库就可以了

七、在IDEA中使用Git

1.在idea中配置git

- 1.如果Git安装在默认路径下，那么idea会自动找到git的位置，如果更改了Git的安装位置则需要手动配置下Git的路径。
- 2.选择File→Settings打开设置窗口，找到Version Control下的git选项就是idea自动找到的git安装目录
- 3.选择git的安装目录后可以点击“Test”按钮测试是否正确配置

2.将工程添加到git

- 1.在idea中创建一个工程，例如创建一个java工程，名称为idea-git-test

2.创建本地仓库：在菜单选择“vcs”→Import into Version Control→Create Git Repository..., 选择工程所在的上级目录。，然后点击“OK”按钮

3.此目录中的工程就可以添加到本地仓库中。也就是可以把idea-git-test工程添加到本地仓库中。选择之后在idea上方的工具栏上就多出了git相关工具按钮

4.将工程添加到本地仓库：直接点击commit按钮，然后在Commit Message中填写日志信息，最后单击下方的commit按钮即可

5.推送到远程：在github上创建一个仓库然后将本地仓库推送到远程。在工程上点击右键，选择git→Repository→push，或者在菜单中选择vcs→git→push

6.点击“Defineremote”链接，配置https形式的URL，git形式的无法通过。然后点击OK

7.点击“push”按钮就讲本地仓库推送到远程，如果是第一次配置推送需要输入github的用户名和密码

3.从远程仓库克隆至idea本地

1.关闭工程后，在idea的欢迎页上有“Check out from version control”下拉框，选择git

2.此处仍然推荐使用https形式的url，点击“test”按钮后显示连接成功。点击OK按钮后根据提示将远程仓库克隆下来，然后倒入到idea中。

4.从服务端拉取代码

如果需要从服务端同步代码可以使用工具条中的“update”按钮