

Git

前言

计划：先了解基本的概念与操作，然后在虚拟机Linux中部署，在主机windows中使用，模拟实战。

学习资料：

[菜鸟教程](#)

[廖雪峰官网](#)

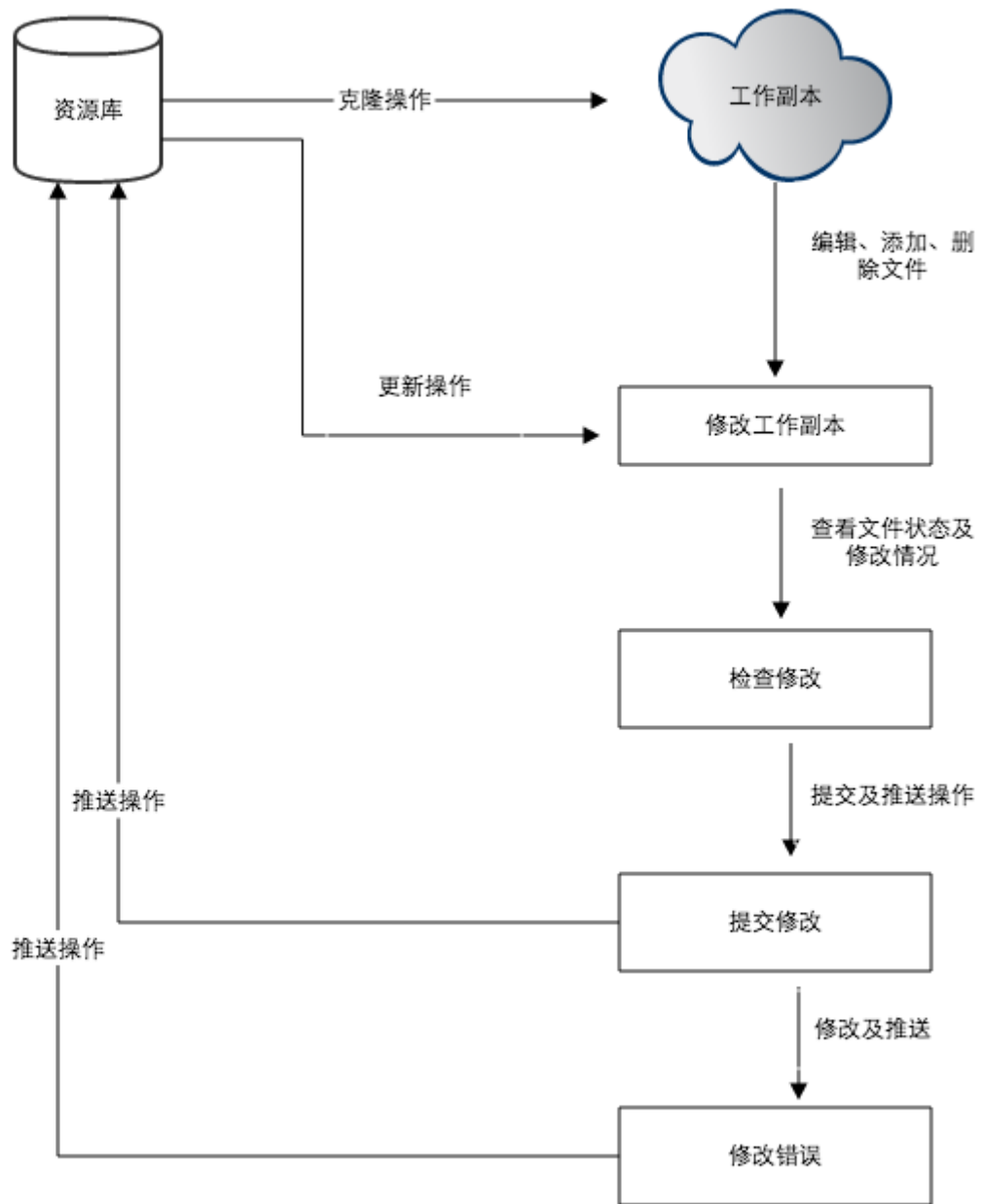
[自信搜索](#)

简单介绍

Git是一个分布式的项目版本管理系统，说白了就是一个管理项目版本的工具。

Git的工作流程

Git 工作流程

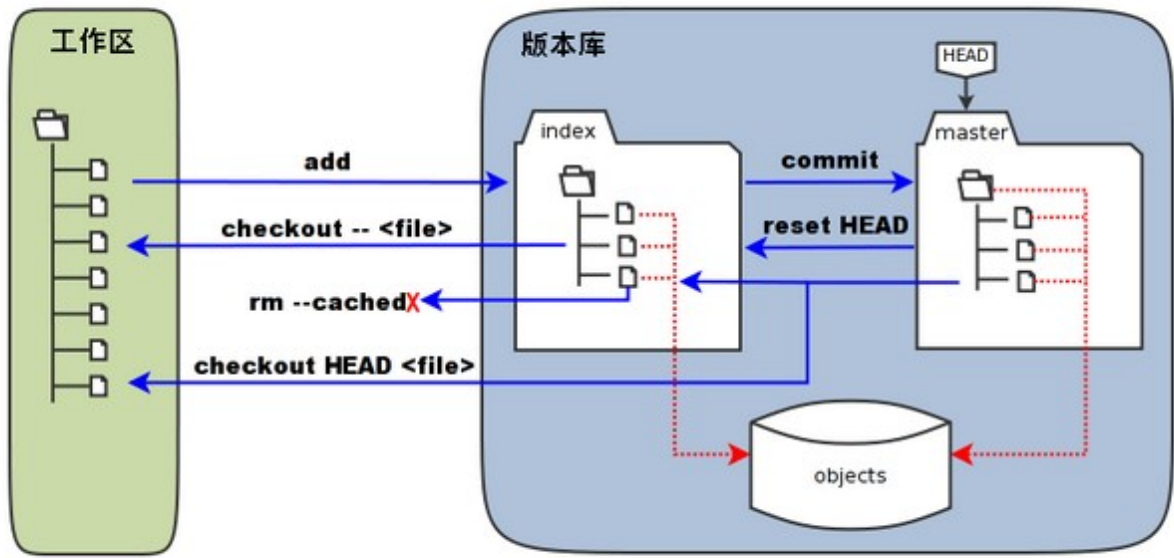
菜鸟教程: <http://www.runoob.com>

工作区、缓存区和版本库的概念

三个东西的概念：

- **工作区**：电脑内能看见的目录。
- **暂存区**：一般存放在 `.git` 目录下的 `index` 文件中，所以有时候也叫索引。
- **版本库**：工作区内的隐藏目录 `.git`（一般看不见）。

三个东西之间的关系：



基本操作命令分类及列表

创建仓库命令

命令	说明
<code>git init</code>	初始化仓库
<code>git clone</code>	拷贝一份远程仓库，也就是下载一个项目

提交与修改命令

命令	说明
<code>git add</code>	添加文件到仓库
<code>git status</code>	查看仓库当前的状态，显示有变更的文件
<code>git diff</code>	比较文件的不同，即暂存区和工作区的差异
<code>git commit</code>	提交暂存区到本地仓库
<code>git reset</code>	回退版本
<code>git rm</code>	删除工作区文件
<code>git mv</code>	移动或重命名工作区文件

提交日志命令

命令	说明
----	----

命令	说明
<code>git log</code>	查看历史提交记录
<code>git blame</code>	以列表形式查看指定文件的历史修改记录

远程操作命令

命令	说明
<code>git remote</code>	远程仓库操作
<code>git fetch</code>	从远程获取代码库
<code>git pull</code>	下载远程代码并合并
<code>git push</code>	上传远程代码并合并

分支管理命令

命令	说明
<code>git branch</code>	列出分支
<code>git branch 新分支名</code>	创建分支
<code>git branch -d 分支名</code>	删除分支
<code>git checkout 分支名</code>	切换分支
<code>git checkout -b 新分支名</code>	同时创建切换分支的快捷方式
<code>git branch -m 旧分支名 新分支名</code>	重命名分支
<code>git merge 目标分支名</code>	分支合并

命令更详细的使用方法见

创建仓库

1. `git init`: 使用当前目录作为Git仓库。
2. `git init 目录名`: 使用目标目录作为Git仓库。

从仓库拷贝（克隆）项目

1. `git clone 被拷贝仓库地址`: 拷贝仓库中的项目到当前目录中（当前目录变成新的仓库）。
2. `git clone 被拷贝仓库地址 指定目录地址`: 拷贝仓库中的项目到指定目录中。

仓库的配置（部分）

查看当前git配置信息

- `git config --list`。

编辑git配置文件：

1. `git config -e` 针对当前仓库
2. `git config -e --global` 针对系统上所有仓库

设置提交代码时的用户信息: `git config --global user.name "名字"` `git config --global user.email 邮箱地址`

注：去掉--global参数则只对当前仓库有效

从工作区添加文件到暂存区

1. `git add 文件名 文件名 ...`：添加若干个文件到暂存区。
2. `git add 目录`：添加指定目录到暂存区，包括子目录。
3. `git add .`：添加当前目录下的所有文件到暂存区。

查看仓库当前状态

1. `git status`：查看上次提交之后是否对文件进行了修改。
2. `git status -s`：简短输出结果

比较文件的不同

`git diff 参数 参数 ...`

参数：

1. 不填参数：查看工作区与缓存区所有文件的不同。
2. 文件名：查看工作区与缓存区中目标文件的不同。
3. `--cached`或`--staged`：查看缓存区中现在的文件和上次提交的文件的不同。
4. `HEAD`：查看工作区与版本库上个版本的差别。[其他版本格式](#)。
5. `--stat`：简要显示结果。

[详细](#)

提交暂存区到本地仓库（版本库）

1. `git commit -m "备注信息"`：提交缓存到版本库中。（备注信息windows用双引号，linux用单引号）
2. `git commit 文件名 文件名 ... -m "备注信息"`：提交缓存区的指定文件到版本库。
3. `git commit -a`：无需手动执行git add命令，直接提交。

注：如果你没有设置 -m 选项，Git 会尝试为你打开一个编辑器以填写提交信息。

回退版本

git reset 参数 版本参数

参数:

1. `--soft`: 回退到某个版本。
2. `--mixed`: 默认, 可以不写。用于重置暂存区的文件与目标版本那次的commit版本保持一致, 工作区内内容保持不变。
3. `--hard`: 销毁工作区中所有未提交的修改内容, 将暂存区与工作区都回到目标版本, 并删除之前的所有信息提交。

注: -hard会删除回退点之前的所有信息

版本参数:

版本	表示方法1	表示方法2
当前版本	HEAD	HEAD~0
上一个版本	HEAD^	HEAD~1
上上一个版本	HEAD^^	HEAD~2
上上上一个版本	HEAD^^^	HEAD~3

以此类推...

删除文件

git rm 参数 文件名

参数:

1. **不填参数**: 从暂存区和工作区删除文件。
2. `-f`: 若文件修改过且已经放入缓存区则需要强制删除。
3. `--cached`: 将文件从暂存区删除, 保留在当前工作目录中。

递归删除: `git rm -r *`

删除当前目录下的所有文件及其子目录

移动或重命名文件

git mv 参数 旧文件名 新文件名

参数:

1. `-f`: 强制重命名。
2. `-n`: 显示重命名发生的改变, 不会进行重命名。

查看日志提交历史记录

`git log`: 查看历史提交记录。

详细请参考[这个](#)和[这个](#)

查看文件历史修改记录

`git blame` 文件名: 以列表形式查看目标文件历史修改记录。

远程仓库的操作

1. `git remote -v`: 显示所有远程仓库。
2. `git remote show` 远程仓库地址: 显示目标远程仓库的信息。
3. `git remote add` 本地仓库 远程仓库地址: 添加远程仓库
4. `git remote rm` 远程仓库名: 删除远程仓库
5. `git remote rename` 旧名 新名: 修改远程仓库名

获取远程代码库

`git fetch` 远程版本库: 下载远程版仓库

下载远程代码并合并

`git pull` 远程主机名 远程分支名:本地分支名: 取回远程主机某个分支的更新, 再与本地的指定分支合并。

上传远程代码并合并

`git push` 远程主机名 本地分支名:远程分支名: 将本地分支的更新, 推送到远程主机。

列出分支

`git branch`: 列出本地分支。

创建分支

`git branch` 新分支名: 从现有分支创建新分支。

删除分支

`git branch -d` 分支名: 删除目标分支。

切换分支

`git checkout` 分支名: 切换到目标分支。v

同时创建切换分支的快捷方式

`git checkout -b 新分支名`：创建一个新分支，并立刻切换到新分支。

重命名分支

`git branch -m 旧分支名 新分支名`：旧分支重命名为新分支名。

分支合并

`git merge 目标分支名`：将目标分支名与当前分支合并。

合并时的冲突即处理

两个分支对文件的修改内容不同，当两分支合并时就会产生冲突，冲突信息会在合并语句执行后显示，对应冲突文件会有冲突提示，这时只需要修改对应的冲突文件，并执行`git add`命令告诉Git冲突已解决即可。

实践所遇问题及解决方法

主机获取虚拟机仓库命令

`git clone 用户名@ip:仓库地址`

主机传输仓库到虚拟机

`git push`

遇到困难：执行命令后报错

```
Users\HS\Desktop\WSQ_Study\Git\Storage\MyGit>git push
192.168.1.108's password:
merging objects: 5, done.
packing objects: 100% (5/5), done.
delta compression using up to 4 threads
compressing objects: 100% (2/2), done.
writing objects: 100% (3/3), 315 bytes | 315.00 KiB/s, done.
total 3 (delta 0), reused 0 (delta 0), pack-reused 0
note: error: refusing to update checked out branch: refs/heads/master
note: error: By default, updating the current branch in a non-bare repository
note: error: is denied, because it will make the index and work tree inconsistent
note: error: with what you pushed, and will require 'git reset --hard' to match
note: error: the work tree to HEAD.
note: error:
note: error: You can set 'receive.denyCurrentBranch' configuration variable to
note: error: 'ignore' or 'warn' in the remote repository to allow pushing into
note: error: its current branch; however, this is not recommended unless you
note: error: arranged to update its work tree to match what you pushed in some
note: error: other way.
note: error:
note: error: To squelch this message and still keep the default behaviour, set
note: error: 'receive.denyCurrentBranch' configuration variable to 'refuse'.
192.168.1.108:/home/ngsiukei/Study/MyGit/.git
[remote rejected] master -> master (branch is currently checked out)
error: failed to push some refs to '192.168.1.108:/home/ngsiukei/Study/MyGit/.git'
```

原因：虚拟机库设置问题

解决方法：进入虚拟机仓库目录下.git目录，修改config，添加图中框住的内容

```
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true

[receive]
    denyCurrentBranch = ignore
```