

git

版本控制

分类

1. 本地版本控制

适合个人

2. 集中版本控制

版本数据放在**服务器**上。

缺点：

- (1) 存在服务器损坏风险
- (2) 无分支

3. 分布式版本控制

- (1) 同步到每个用户本地
- (2) 可离线提交

安装配置+基础知识学习 (Linux基础命令)

Git Bash : Unix与Linux风格的命令行, 使用最多, 推荐最多

Git CMD : Windows风格的命令行

Git GUI : 图形界面的Git, 不建议初学者使用, 尽量先熟悉常用命令

linux基本指令

基本的Linux命令学习

- 1)、cd : 改变目录。
- 2)、cd .. 回退到上一个目录, 直接cd进入默认目录
- 3)、pwd : 显示当前所在的目录路径。
- 4)、ls(ll) : 都是列出当前目录中的所有文件, 只不过ll(两个l)列出的内容更为详细。
- 5)、touch : 新建一个文件 如 touch index.js 就会在当前目录下新建一个index.js文件。
- 6)、rm : 删除一个文件, rm index.js 就会把index.js文件删除。
- 7)、mkdir : 新建一个目录, 就是新建一个文件夹。
- 8)、rm -r : 删除一个文件夹, rm -r src 删除src目录
- 9)、mv 移动文件, mv index.html src index.html 是我们要移动的文件, src 是目标文件夹, 当然, 这样写, 文件夹在同一目录下。
- 10)、reset 重新初始化终端/清屏。
- 11)、clear 清屏。
- 12)、history 查看命令历史。
- 13)、help 帮助。
- 14)、exit 退出。
- 15)、#表示注释



切勿在Linux上尝试

- 1 | rm -rf /
- 2 | 会格式化电脑

配置

配置环境变量只是为了可以随时使用，但已存在于右键，无需配置，且已自动配置

1. 查看

- 查看配置: **git config -l**
- 查看系统配置:**git config --system --list**
- 查看本地配置:**git config --global --list**

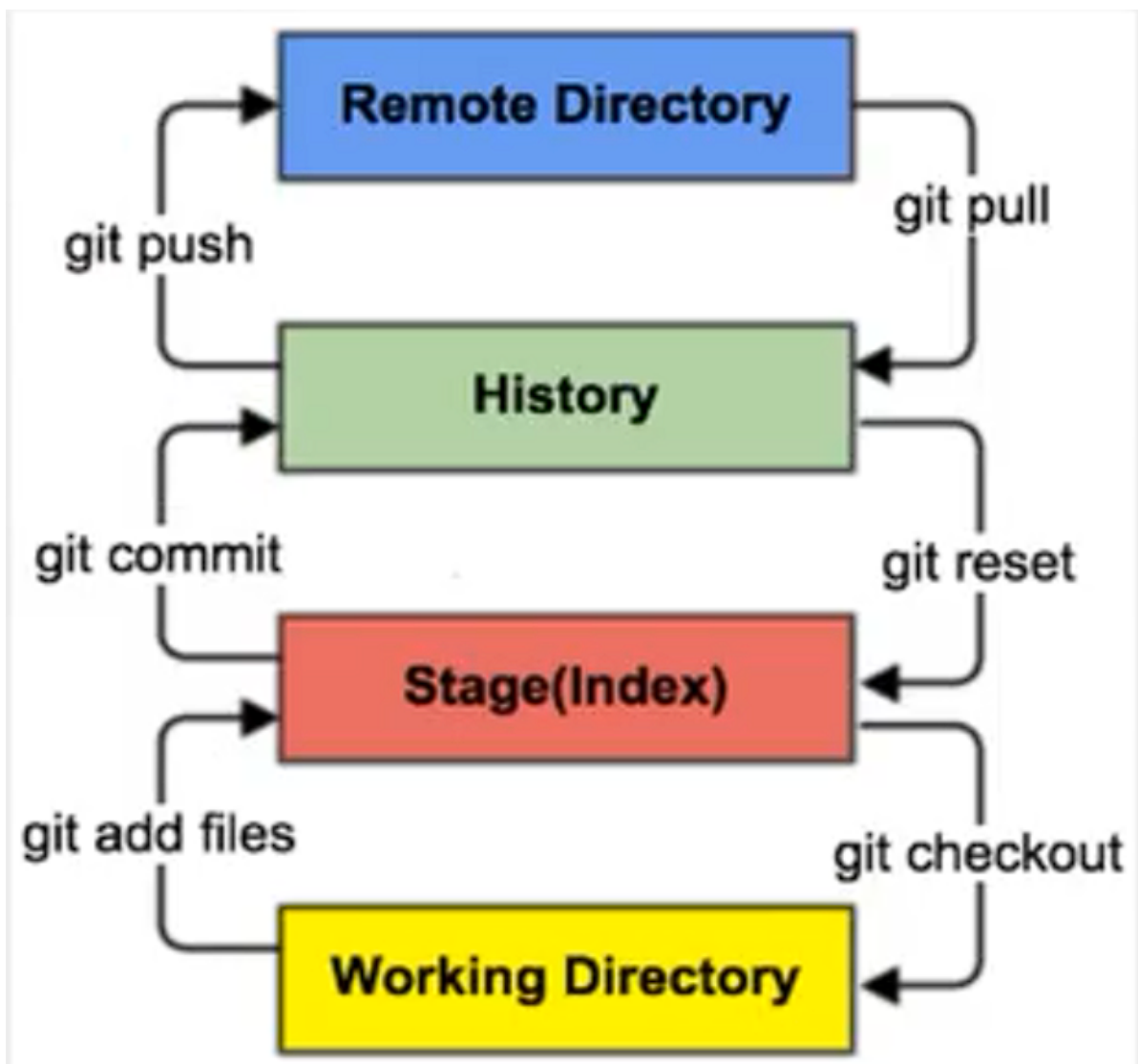
2. 配置(本质读写文件)

- git config --global user.name "Lancaster"
- git config --global user.email LancasterLiu_cryp@163.com

基本理论

工作区域

1. 工作目录
2. 暂存区
3. 资源库
4. git远程仓库



- 工作区：本地存放项目文件代码
- 暂存区：临时存放改动，其实为一个列表信息文件
- 本地仓库：存放文件变化数据
- 远程仓库：git

搭建本地项目实践

1. 在本地搭建一个仓库

- 1 #初始化
- 2 `git init`

```
1 #克隆远程仓库
2 git clone [url]
3 #url:https://github.com/LancasterLiu/spider
```

文件操作

文件4种状态：

1. Untracked（未跟踪）：即在文件夹中，但不在git库中。可通过 `git add` 使状态变为**Staged**
2. Unmodify（文件已入库但未修改）：可通过 `git rm` 使状态变为**Untracked**即移除；或修改文件使状态变为**Modified**
3. Modified（已修改）：可通过 `git add` 使状态变为**Staged**；或可通过 `git checkout` 使状态变为**Unmodify**，即还原原来文件
4. Staged（暂存）：可通过 `git commit` 使修改同步到git库，状态变为**Unmodify**；或通过 `git reset HEAD filename` 取消暂存，状态变为**Modify**

查看状态

```
1 #查看所有文件状态
2 git status
```

```
1 #查看指定文件状态
2 git status "filename"
```

```
1 #添加所有文件到暂存区
2 git add .
```

```
1 #提交暂存区文件到git仓库
2 git commit -m "文件描述"
```

上传时忽略文件

在主目录下创建.gitignore文件，可以控制不上传哪些文件

- 使用手册
 - #为注释
 - 可使用linux通配符
 - ! 代表例外规则，不被忽略
 - /路径分隔符
- 举例

```
1 #注释
2 *.txt          #忽略所有txt文件
3 !lib.txt       #lib.txt除外
4 /temp          #不忽略temp
5 main/          #忽略main下所有文件
6 main/*.txt     #忽略main下所有txt文件
```

配置SSH公钥

1. 生成公钥文件

```
1 #直接在bash里面输入  
2 ssh-keygen -t rsa
```

2. 复制.pub文件信息至gitee

3. 创建git远程仓库

分支

- 查询

```
1 #本地分支  
2 git branch  
3 #远程分支  
4 git branch -r
```

- 创建

```
1 #仍停留在原分支  
2 git branch 分支名  
3 #切换到新分支  
4 git checkout -b 分支名
```

- 合并

```
1 | #合并指定分支到当前分支
2 | git merge 分支名
```

- 删除

```
1 | #本地
2 | git branch -d 分支名
3 | #远程
4 | git branch -dr 分支名
5 | git push origin --delete 分支名
```

- 切换

```
1 | git checkout 分支名
```