



# Git版本控制

## 1. 版本控制

### 1.1 优势

1. 协同开发
2. 追踪历史记录
3. 统计工作量
4. 等等

### 1.2 本地版本控制

本地运行和操作，繁琐

### 1.3 集中版本控制（SVN）

运行在中央服务器上

较依赖服务器

必须联网才能使用

### 1.4 分布式版本控制（Git）

每个人都拥有全部的代码

Git是目前最先进的分布式版本控制工具

## 2. Git历史

Linux内核开源项目需要使用到BitKeeper工具（2002）

BitKeeper开发公司拒绝与Linux合作（2005）

Git在两周内被开发出来（2005）

## 3. Git环境配置

### 3.1 Git下载

<https://git-scm.com/downloads>

镜像下载速度快

Git Bash：Linux风格，使用的最多

Git CMD：Windows风格控制

Git GUI：图形界面

### 3.2 基本操作命令

- 基础

cd 改变目录

cd .. 返回上一级

ls 列出当前目录下所有文件

clear 编辑器清屏

pwd 显示当前工作路径

history 查看历史命令

help 帮助

exit 退出

- 文件操作

touch 创建新文件

rm 删除文件

mkdir 创建文件夹

rm -r 删除文件夹

mv .. src .. 移动文件

## 3.3 Git配置

所有的配置文件都保留在本地

```
git config -l
```

 查看配置

```
git config --system -l
```

 查看系统配置

```
git config --global -l
```

 查看本地配置

设置用户名和邮箱

```
git config --global user.name
```

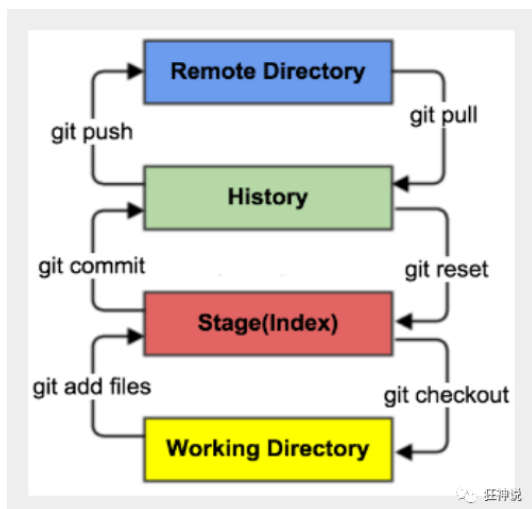
 配置用户名

```
git config --glocal user.email
```

 配置邮箱

## 4. Git基础理论（核心）

### 4.1 工作区域



- **Workspace**: 工作区，就是你平时存放项目代码的地方
- **Index / Stage**: 暂存区，用于临时存放你的改动，事实上它只是一个文件，保存即将提交到文件列表信息
- **Repository**: 仓库区（或本地仓库），就是安全存放数据的位置，这里面有你提交到所有版本的数据。其中HEAD指向最新放入仓库的版本
- **Remote**: 远程仓库，托管代码的服务器，可以简单的认为是你项目组中的一台电脑用于远程数据交换

### 4.2 工作流程

- git管理的文件有三种状态：已修改（modified）,已暂存（staged）,已提交(committed)，git的工作流程一般是这样的：

1. 在工作目录中添加、修改文件；

2. 将需要进行版本管理的文件放入暂存区域；
3. 将暂存区域的文件提交到git仓库。

## 5. Git项目搭建

### 5.1 搭建

本地搭建 `git init`

远程克隆 `git clone [url]`

### 5.2 Git文件操作

查看指定文件状态 `git status [filename]`

查看所有文件状态 `git status`

添加所有文件到暂存区 `git add .`

提交暂存区中的内容到本地仓库，-m 后面跟提交信息 `git commit -m`

### 5.3 忽略文件

有些时候我们不想把某些文件纳入版本控制中，比如数据库文件，临时文件，设计文件等

在主目录下建立".gitignore"文件，此文件有如下规则：

1. 忽略文件中的空行或以井号（#）开始的行将会被忽略。
2. 可以使用Linux通配符。例如：星号（\*）代表任意多个字符，问号（?）代表一个字符，方括号（[abc]）代表可选字符范围，大括号（{string1,string2,...}）代表可选的字符串等。
3. 如果名称的最前面有一个感叹号（!），表示例外规则，将不被忽略。
4. 如果名称的最前面是一个路径分隔符（/），表示要忽略的文件在此目录下，而子目录中的文件不忽略。
5. 如果名称的最后面是一个路径分隔符（/），表示要忽略的是此目录下该名称的子目录，而非文件（默认文件或目录都忽略）。

```
1 #为注释
2 *.txt      #忽略所有 .txt 结尾的文件, 这样的话上传就不会被选中!
3 !lib.txt   #但lib.txt 除外
4 /temp      #仅忽略项目根目录下的TODO文件, 不包括其它目录temp
5 build/     #忽略build/ 目录下的所有文件
6 doc/*.txt  #会忽略 doc/notes.txt 但不包括 doc/server/arch.txt
```

## 6. 其他

- 网络原因，有时会出现push失败的情况，换个时间push即可
- 提交完github时出现贡献图没有显示的问题，需要保证提交的邮箱和GitHub保存的邮箱一致



参考资料：

B站教程： <https://b23.tv/mANvQ9g>

遇见狂神说笔记： <https://mp.weixin.qq.com/s/Bf7uVhGiu47uOELjmC5uXQ>