

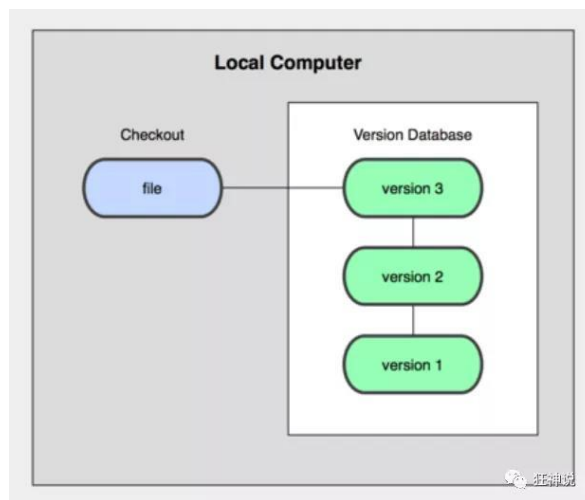
1 git 是干什么的？

简单来讲：版本控制。

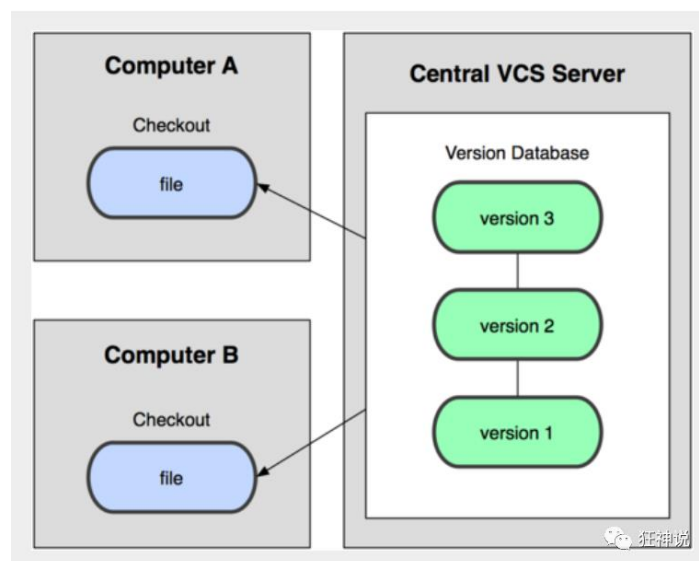
具体来讲：在开发的过程中用于管理我们对文件、目录或工程等内容的修改历史，方便查看更改历史记录，备份以便恢复以前的版本。

2 为什么是分布式的版本控制？

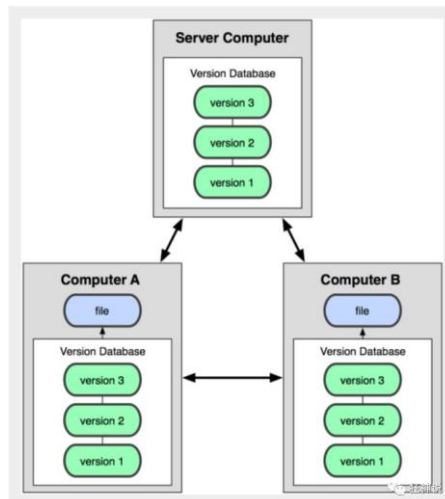
1) 本地版本控制



2) 集中版本控制 SVN



3) 分布式版本控制 Git



分析优劣：

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，而工作的时候，用的都是自己的电脑，所以首先要从中央服务器得到最新的版本，然后工作，完成工作后，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，对网络带宽要求较高。

Git 是分布式版本控制系统，没有中央服务器，**每个人的电脑就是一个完整的版本库**，工作的时候不需要联网了，因为版本都在自己电脑上。协同的方法是这样的：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。Git 可以直接看到更新了哪些代码和文件！

3 Git 的安装与配置

下载：1) 官网 2) 淘宝镜像：<http://npm.taobao.org/mirrors/git-for-windows/>

安装：无脑下一步即可！安装完毕就可以使用了！

配置：所有的配置文件，其实都保存在本地！

- system 系统级配置在 Git 安装目录下的 gitconfig 中。
- global（当前用户）的配置在 C:\Users\Administrator\.gitconfig 中

代码测试：

1) 查看配置 `git config -l`

2) 查看系统配置 `git config --system --list`

3) 查看当前用户（global）配置 `git config --global --list`

当你安装 Git 后首先要做的事情是设置你的用户名称和 e-mail 地址。这是非常重要的，因为每次 Git 提交都会使用该信息。它被永远的嵌入到了你的提交中：

- `git config --global user.name "kuangshen" #名称`
- `git config --global user.email 24736743@qq.com #邮箱`

4 远程服务器免密码登录

进入 C:\Users\Administrator\.ssh 目录下，生成公钥：`ssh-keygen`。将公钥信息 public key 添加到 github 账户中即可！

5 Git 的项目搭建

创建本地仓库的方法有两种：一种是创建全新的仓库，另一种是克隆远程仓库。

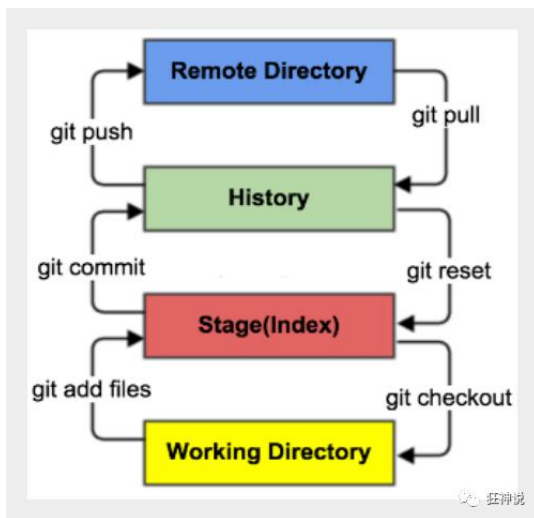
1) **创建全新的仓库**，需要用 GIT 管理的项目的根目录执行：`git init`。

执行后可以看到，仅仅在项目目录多出了一个.git 目录，关于版本等的信息都在这个目录里面。

2) **克隆远程仓库**

```
git clone [url] #https://gitee.com/kuangstudy/openclass.git
```

6 提交到服务器（github）



- 1) 查看指定文件状态 `git status [filename]`
- 2) 查看所有文件状态 `git status`
- 3) 添加所有文件到暂存区 `git add .`
- 4) 提交暂存区中的内容到本地仓库 `git commit -m "消息内容"` （忽略文件.gitignore）
- 5) 将本地仓库推送到服务器。 `git push`

7 时空穿梭

HEAD 指向的版本就是当前版本，因此，Git 允许我们在版本的历史之间穿梭，使用命令 `git reset --hard commit_id`。

- 穿梭前，用 `git log` 可以查看提交历史，以便确定要回退到哪个版本。
- 要重返未来，用 `git reflog` 查看命令历史，以便确定要回到未来的哪个版本。

8 删除文件

一般情况下，你通常直接在文件管理器中把没用的文件删了，或者用 `rm` 命令删了。这个时候，Git 知道你删除了文件，因此，工作区和版本库就不一致了，`git status` 命令会立刻告诉你哪些文件被删除了。

现在你有两个选择，一是确实要从版本库中删除该文件，那就用命令 `git rm` 删掉，并且 `git commit`。另一种情况是删错了，因为版本库里还有呢，所以可以用：`git checkout` 很轻松地把误删的文件恢复到最新版本，其实是用版本库里的版本替换工作区的版本，无论工作区是修改还是删除，都可以“一键还原”。

9 强制覆盖本地代码（与 git 远程仓库保持一致）

`git fetch --all`

```
git reset --hard origin/master  
git pull
```

第一个是：拉取所有更新，不同步；

第二个是：本地代码同步线上最新版本(会覆盖本地所有与远程仓库上同名的文件)；

第三个是：再更新一次（其实也可以不用，第二步命令做过了其实）

注：所有的版本控制系统，其实只能跟踪文本文件的改动，比如 TXT 文件，网页，所有的程序代码等等，Git 也不例外。

参考：

<https://www.liaoxuefeng.com/wiki/896043488029600>

<https://www.bilibili.com/video/BV1FE411P7B3?p=1>

<https://mp.weixin.qq.com/s/Bf7uVhGiu47uOELjmC5uXQ>