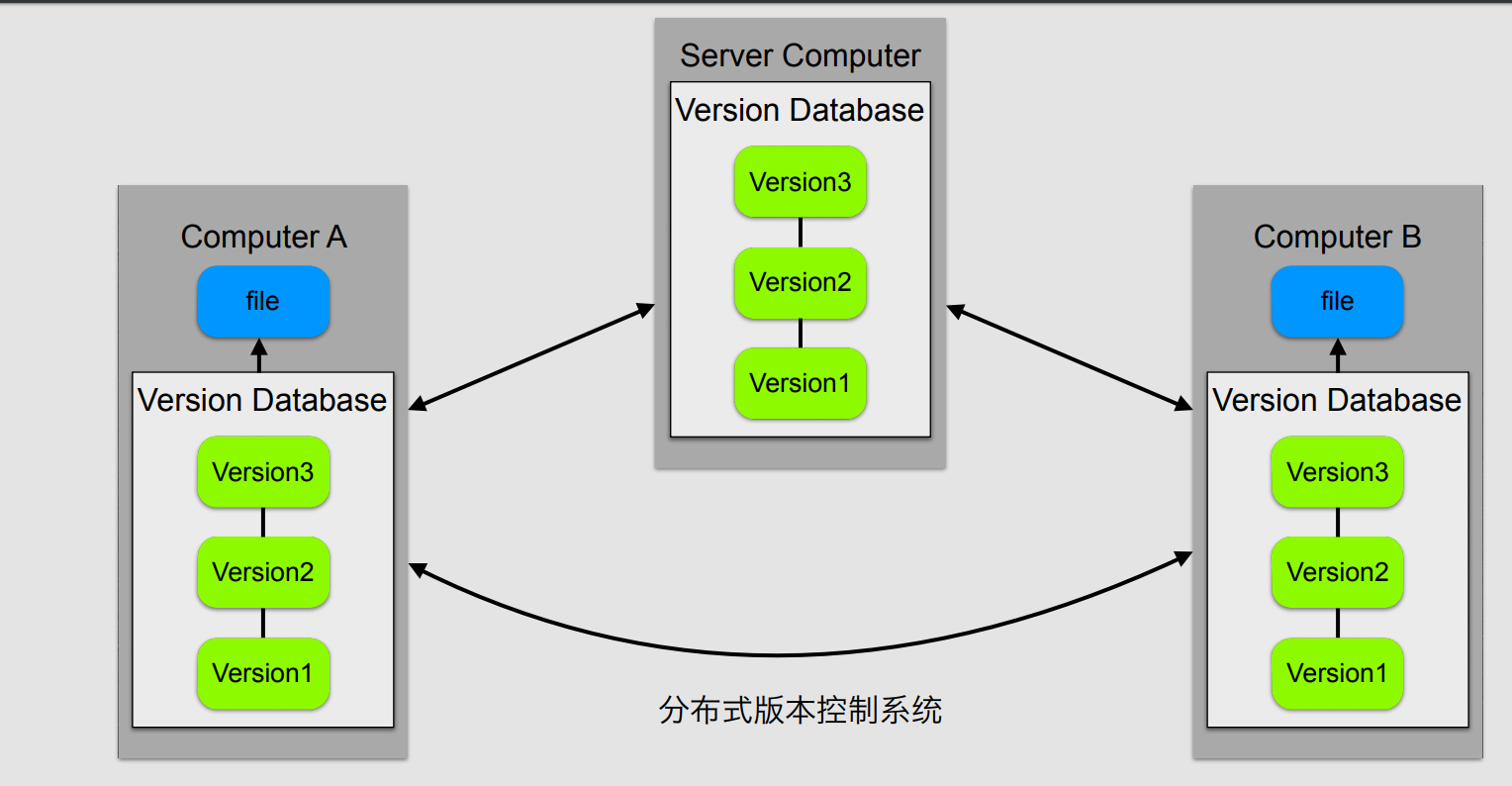
**Git：分布式版本控制系统**

Git是分布式版本控制系统，那么它就没有中央服务器的，每个人的电脑就是一个完整的版本库，这样，工作的时候就不需要联网了，因为版本都是在自己的电脑上。既然每个人的电脑都有一个完整的版本库，那多个人如何协作呢？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。

下图就是分布式版本控制工具管理方式：



仓库（版本库）：相当于一个专门用来存放代码的目录。这个目录里面的所有文件都可以Git管理，每个文件的增删改查都能被Git跟踪到

**Git的结构：**

**git工作流程**

一般工作流程如下：

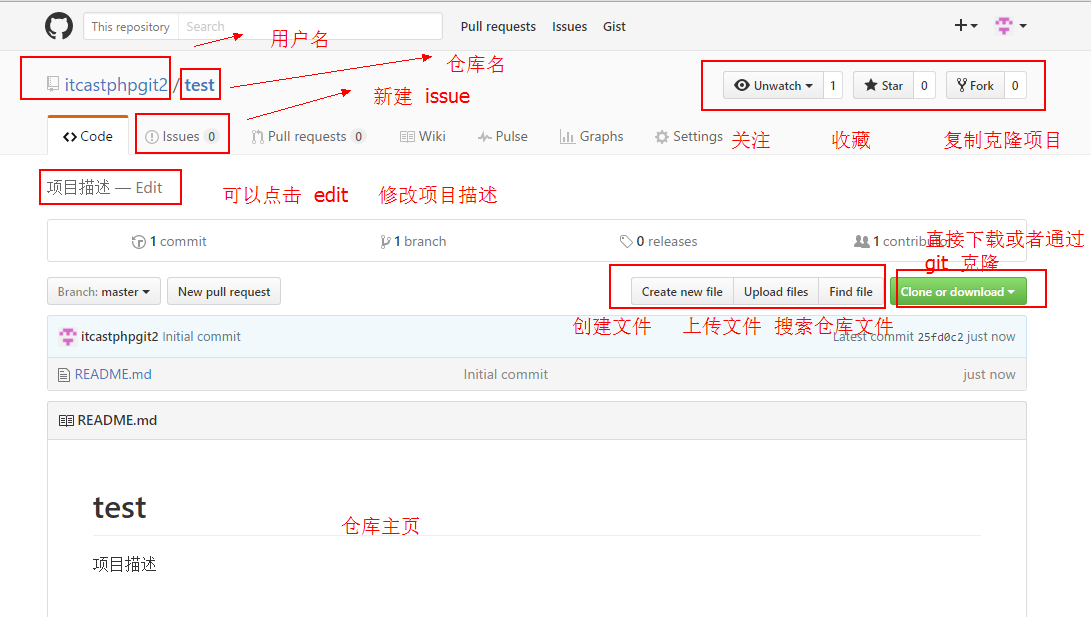
1. 从远程仓库中克隆 Git 资源作为本地仓库。
2. 从本地仓库中checkout代码然后进行代码修改
3. 在提交前先将代码提交到暂存区。
4. 提交修改。提交到本地仓库。本地仓库中保存修改的各个历史版本。
5. 在修改完成后，需要和团队成员共享代码时，可以将代码push到远程仓库。

下图展示了 Git 的工作流程：



**Github:开源社区**

**Github线上仓库结构介绍：**

**Github Issues**

作用：发现代码BUG，但是目前没有成型代码，需要讨论时用；或者使用开源项目出现问题时使用

**fork与git clone的区别**

1.区别

git clone 是在自己电脑（这里我是ubuntu）直接敲命令，结果是将github仓库中的项目克隆到自己本地电脑中了

fork是直接访问github网站，在项目页面中点击fork，然后自己github项目中就会多出一个复制的项目

2.用法

如果我们想要修改他人github项目的话，我们直接git clone代码到本地是不能pull的，所以我们使用fork，先把代码复制到自己的github仓库，然后git clone到本地修改，然后在提交pull（这里的pull是pull到自己github仓库了，我们自己的github仓库中的代码是fork源的一个分支），这时候我们想要把修改的代码提交给他人的话，就可以在自己github上pull，等其他人看到后就可以把代码做一个合并