git add命令和git commit命令执行的时候，背后的原理是什么？带出来git体系结构853769620

1、git本地仓库结构

git项目有3个主要的部分组成：工作区（working directory / working tree），暂存区（staging area），版本库（git directory / repository）

git directory / repository：git版本库，其实就是git用于存储自己的元数据，以及文档数据库的地方，默认就是在项目的.git隐藏目录中

working directory / working tree：工作区，保存的是一个项目当前的一个版本对应的所有文件，这些文件是从git版本库中的压缩后的数据库中提取出来，然后放到我们的磁盘上去。

staging area：暂存区，就是一个文件，包含在git版本库中，主要是保存了下一次要提交到的那些文件信息。在git中，对暂存区有另外一个名称，叫做index，也就是索引。

上面三个区域的协作关系大致如下：

（1）首先会在工作区修改某个版本的文件

（2）将某些修改后的文件放入git暂存区中，准备下一次提交到git版本库中去

（3）执行一个提交操作，将暂存区中的文件保作为一个快照保存到git版本库中去

如果一个文件，已经有一个版本被保存到了版本库，那么就是committed状态；如果这个文件被修改了，同时被加入了暂存区，那么就是staged状态；如果这个文件修改了，还没有加入暂存区，那么就是modified状态。

工作区，working directory

所谓的工作区，指的就是当前你的git管理的项目，在本地的那个目录，也就是你能直接看到，编辑的那个目录，这就是工作区。

git add和git commit两个命令幕后的原理

2、git的重要机制

Git有很多特有的机制，都是跟普通的版本控制系统不一样的

（1）快照机制

普通的版本控制系统，比如说CVS，SVN等，是通过一开始提交一个原始文件，然后后面每次对文件进行修改之后再次提交，都维护这次提交对应的一个差异，通过维护每个版本的差异，就可以通过应用差异，或者回退差异，来前进或者后退文件的版本。

Git用的不是这种维护每次提交的差异，而是用的快照。每次提交文件，都是保存一份这个文件当前这个状态的一个完整快照，同时对这次提交维护一个指针，指向这个文件快照。

（2）本地化操作

大多数的git版本控制操作，只要在本地执行即可，所有的版本文件都在本地，因此操作是非常快速的。相比较于那些依赖网络的集中式版本控制系统来说，他们的大多数操作要依赖网络，速度是很慢的。

比如说通过git查看提交历史，比较历史文件的差异，都可以在本地完成，不需要通过服务器做任何事情。

如果我们在飞机或者或者上，没有网；或者在家里，没有vpn。都没有问题，随便做开发、写代码，提交代码，在本地就可以了，等有网络的时候，再把提交的版本推送到远程服务器上去。但是SVN之类的，就不可以提交了，因为没有网，没法连接到服务器。

（3）完整性保证

git在存储任何文件之前，都会对其执行一个校验和，然后用校验和指向那个文件。这是git内核保证的，这样我们是不可以手工修改git版本库中的任何文件的，因为修改了文件之后，会导致计算出来的校验和与之前保存的校验和不匹配，文件会破损。

git用的是SHA-1 hash算法来计算校验和，这是一个40位的字符串，基于文件的内容计算出来的，看起来大概是这样的：

24b9da6552252987aa493b52f8696cd6d3b00373

如果手动破坏.git中存储的文件的内容，git会不承认，因为内容变化之后，会导致内容计算出来的SHA-1 40位的hash值变化，跟之前存储的hash值不同，就认为文件破损

（4）仅仅添加数据

git通常来说，仅仅会在自己的数据库中添加数据，因此提交文件到git之后，很少会丢失，而且如果我们定期提交文件到远程服务器，就更少丢失。