**Git原理 - Git对象总结**

**1、Git操作最基本的流程**

**1）创建工作目录对工作目录进行修改**。

**2）执行git add ./命令添加文件到暂存区**

相当于执行了如下两个底层命令：

* git hash-object -w文件名（修改了多少个工作目录中的文件此命令就要被执行多少次）
* git update-index

说明：git add命令做了什么事情？

表面上是将工作目录中的文件添加到暂存区中，其实真正的流程是：

* 先将工作目录中的文件，生成blob对象存储到本地版本库中，
* 一个文件生成一个blob对象，一个文件执行一次git hash-object -w 文件路径命令。
* 再通过git update-index命令，把本地版本库中blob对象，生成文件的索引（快照），存储到暂存区中。

|  |
| --- |
| 所以说Git是绝对安全的，只要你对文件做过的修改，哪怕没有提交到本地版本库，只是提交到暂存区，Git也会帮你记录下来。 |

**3）执行git commit -m "注释内容"命令，把暂存区的快照提交到本地版本库**

相当于执行了如下两个底层命令：

* git write-tree：生成tree对象。
* git commit-tree：生成commit对象。

说明：git commit命令做了什么事情？

表面上是将暂存区的文件索引提交到了本地版本库中，其实真正的流程是：

* 先通过git write-tree命令，把暂存区中的索引信息，生成一个tree对象存储到本地版本库中。
* 然后通过git commit-tree命令，把上面生成的树对象进行封存，生成一个commit对象，存储到本地版本库中。

**重点提示：一个commit对象肯定会对应一个tree对象（单方向1对1的关系），一个commit对象是不会对应两个tree对象的。（如上说明）**

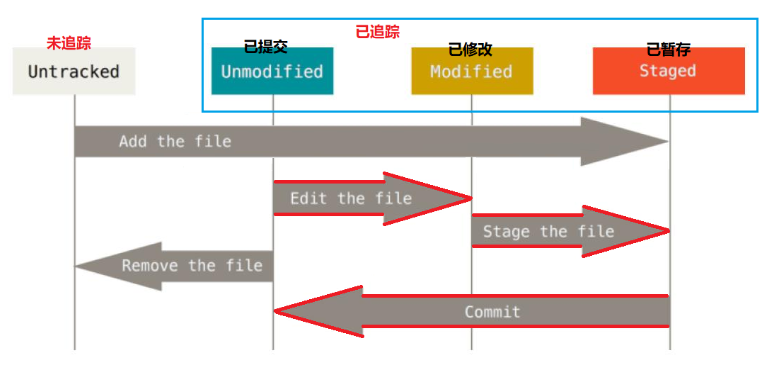
**2、工作目录中文件的状态**

工作目录下面的所有文件都不外乎这两种状态：**已跟踪**状态或**未跟踪**状态。

**已跟踪**的文件是指本来就被纳入版本控制管理的文件，在之前的快照中有它们的记录，工作一段时间后，它们的状态会分为**已提交**状态，**已修改**状态或者**已暂存**状态，这三种。

然后所有其他文件都属于未跟踪文件。它们既没有上次更新时的快照，也不在当前的暂存区域。

使用Git时的文件状态变化周期如下图所示：



|  |
| --- |
| 提示：初次克隆某个仓库到本地时，工作目录中的所有文件都属于已跟踪文件，且状态为已提交；在编辑过某些文件之后，Git将这些文件标为已修改。我们逐步把这些修改过的文件放到暂存区域，直到最后一次性提交所有这些暂存起来的文件。 |

**3、Git效率说明**

我们经历了一次完整的Git提交过程，现在我们来思考一下Git提供的这三种对象带来的高效率：

* 首先Git会对所有的文件内容进行压缩，这使得即使仓库中存储了非常多的内容，而.git也不会很大，
* 然后就是速度，考虑这样的情况，当我们修改了一个文件的时候，Git会去计算这个文件的SHA-1散列值。
* 如果该散列值所得到的路径已经存在，那就说明，这个文件并没有被真正修改（也可以是改了然后又改了回来），这时就不会在本地版本库中存储新的对象。也就是说blob对象跟文件名一点关系都没有，两个不同名字的文件，只要他们的内容相同，在Git的眼里他就是一个blob对象，且只有一份。
* 如果我们真正的修改了一个文件，那么Git会计算这个文件的散列值，然后将这个文件压缩存储在objects目录中。
* 这样设计的可以大大的节约存储的空间，也提升了Git的存储速度。
* 如果我们需要进行一次提交操作，是先对原来的文件进行更改，然后需要创建一个相应的树结构，来记录这些文件的变化。也就是每一次提交都创建一个顶层树对象来表示这次提交快照。
* Git会对比前一个提交的顶层树对象，然后将没有改变的树对象或数据对象直接复制到新创建的这个顶层树对象中，将改变的树对象或数据对象，进行覆盖，最后再提交到本地版本库。

所以说决定你仓库大小的并不是完全在于每个文件的大小，而是你修改提交的次数，修改的次数越多，产生的树对象、数据对象和提交对象也就越多。

|  |
| --- |
| 参考：[https://blog.csdn.net/songyunzu/article/details/92023816](https://links.jianshu.com/go?to=https%3A%2F%2Fblog.csdn.net%2Fsongyunzu%2Farticle%2Fdetails%2F92023816) |

作者：繁华似锦Fighting 链接：https://www.jianshu.com/p/7f06948c0dac 来源：简书 著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。