

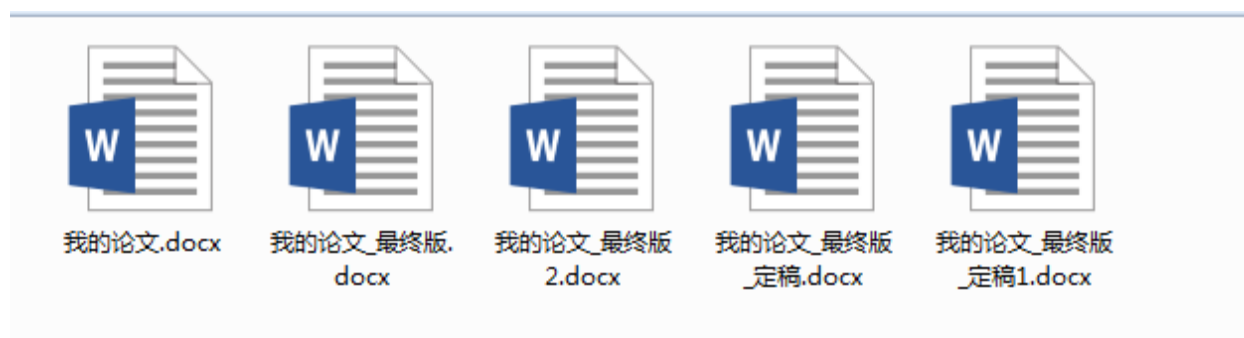
- [Git权威指南中文手册](#)
- [GitHub 团队美女写的一套 Git 使用教程](#)
- [ProGit 中文版](#)
- [廖雪峰的 Git 教程](#)

# 一：Git 简介

Git 是目前世界上最先进的分布式版本控制系统。

## 关于版本控制

相信大家一定有过写论文的经历，写论文的时候大家一定经历了许多的修改，那么最后可能会变成下面这个样子



这么多的文件，想把其他的都删了保留一个最新版本的可是又担心后面需要前面某个版本中被删掉的内容，无奈只有保存，而且有另一种情况，如果是两个人合作写的这篇论文，那么比如另一个人 Copy 了一份文件去写了，然后你继续写你的部分，然后后面合入的话你必须得记得你做了哪些改动，然后把你的改动和他的改动合并起来，是不是感觉非常要命。

这时候你想如果有一个软件可以记录你的文件的每一次改动，还可以多人协作编辑，这样就不用保存一堆类似的文件了，还可以提高多人协作工作的效率。版本控制软件应运而生。

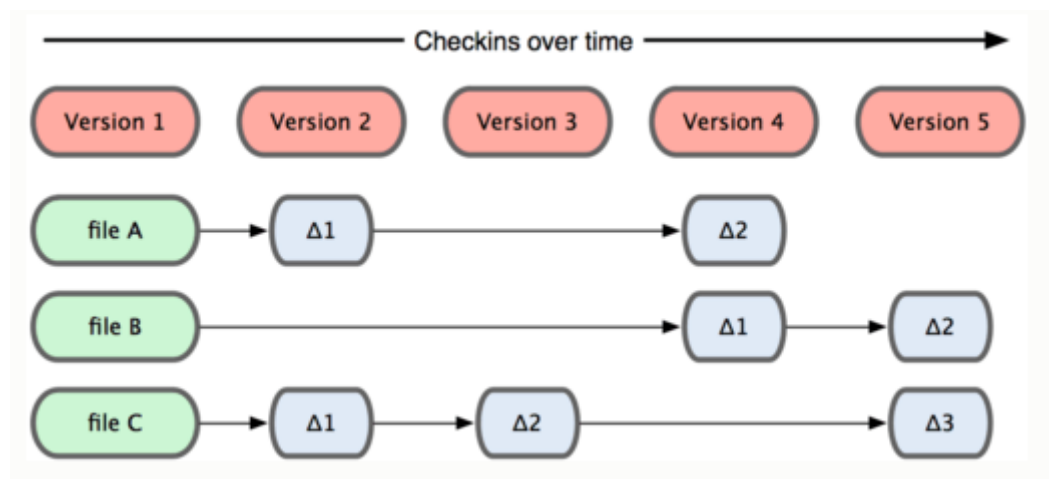
## Git 的来历

大家肯定都听说过 Linux，它是 Linus 在1991 创建的一个开源的操作系统软件，因为是开源的，所以世界上好多的志愿者都自发的参与到 Linux 的功能开发和bug修改中去了，早期的时候志愿者们通过把 diff 文件发送给 Linus，然后他手动的把这些修改合入到 Linux 源码中，不过后来随着代码库越来越大，已经很难手动的再去进行代码库的管理了，大家都对这种手动管理的方式表示了不满，到2002 年，Linux 开源项目组开始启用分布式版本控制系统 BitKeeper 来管理和维护代码。不过到了2005年，由于开发Samba的Andrew试图破解BitKeeper的协议被BitMover公司发现了，于是 BitKeeper 的商业公司同 Linux 内核开源社区的合作关系结束，他们收回了免费使用 BitKeeper 的权力。然后牛人是这样做的，Linus花了两周时间自己用C写了一个分布式版本控制系统，它就是 Git！

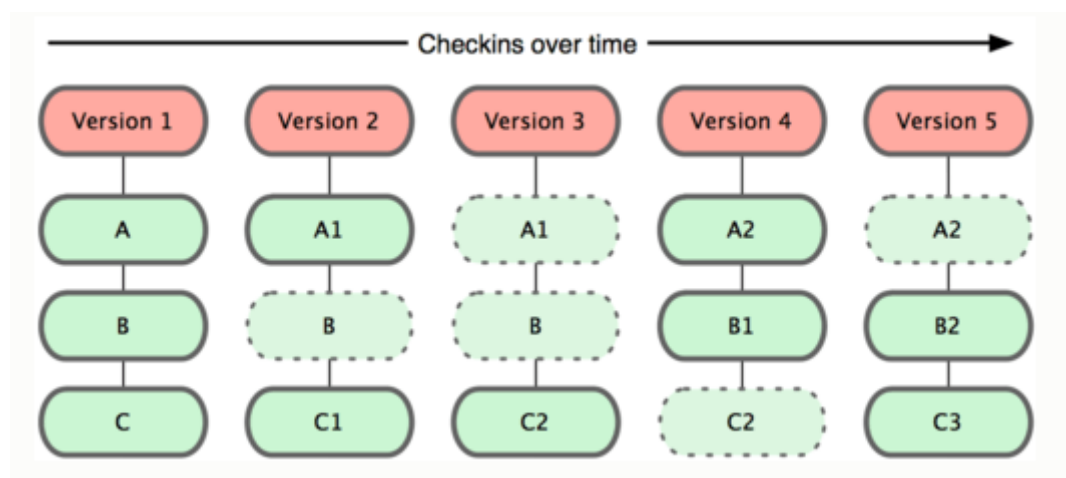
## git 的特点

- **直接记录快照，而非差异比较**

git 与其他版本控制系统的差异在于，git 只关心文件整体是否发生改变，而其他的系统（CVS，Subversion，Perforce，Bazaar 等等）则关心的是文件内容的具体差异，如具体修改了那一行，如下示意图：



Git 并不保存这些前后变化的差异数据。实际上，Git 更像是把变化的文件作快照后，记录在一个微型的文件系统中。每次提交更新时，它会纵览一遍所有文件的指纹信息并对文件作一快照，然后保存一个指向这次快照的索引。为提高性能，若文件没有变化，Git 不会再次保存，而只对上次保存的快照作一链接。Git 的工作方式如下图：



- **近乎所有操作都是本地执行**

由于 Git 是分布式的，每个机器都保存了他的完整的内容和记录，所以 Git 中绝大多数操作都只用访问本地文件和资源，不用联网。但如果用 CVCS（集中式版本管理系统）的话，差不多所有操作都需要连接网络。

例如：如果要浏览项目的历史更新摘要，Git 不用跑到外面的服务器上去取数据回来，而直接从本地数据库读取后展示给你看。所以任何时候你都可以马上翻阅，无需等待。如果想要看当前版本的文件和一个月前的版本之间有何差异，Git 会取出一个月前的快照和当前文件作一次差异运算，而不用请求远程服务器来做这件事，或是把老版本的文件拉到本地来作比较。

用 CVCS 的话，没有网络或者断开 VPN 你就无法做任何事情。但用 Git 的话，就算你在飞机或者火车上，都可以非常愉快地频繁提交更新，等到了有网络的时候再上传到远程仓库。换作其他版本控制系统，这么做几乎不可能或者非常麻烦。比如 Subversion 或 CVS，不联网虽然可以编辑文件，但无法提交更新，因为数据库在网络上。

- **时刻保持数据完整性的SHA-1值 (Commit id)**

在保存到 Git 之前，所有数据都要进行内容的校验和 (checksum) 计算，并将此结果作为数据的唯一标识和索引。换句话说，不可能在你修改了文件或目录之后，Git 一无所知。这项特性作为 Git 的设计哲学，建在整体架构的最底层。所以如果文件在传输时变得不完整，或者磁盘损坏导致文件数据缺失，Git 都能立即察觉。

Git 使用 SHA-1 算法计算数据的校验和，通过对文件的内容或目录的结构计算出一个 SHA-1 哈希值，作为指纹字符串。该字符串由 40 个十六进制字符 (0-9 及 a-f) 组成，看起来就像是：

```
30bf169877a5eaa78ae1ba3de1941ba9f0c53e74
```

Git 的工作完全依赖于这类指纹字符串，所以你会经常看到这样的哈希值。实际上，所有保存在 Git 数据库中的东西都是用此哈希值来作索引的，而不是靠文件名。

## GIT中保证文件完整性的SHA-1值

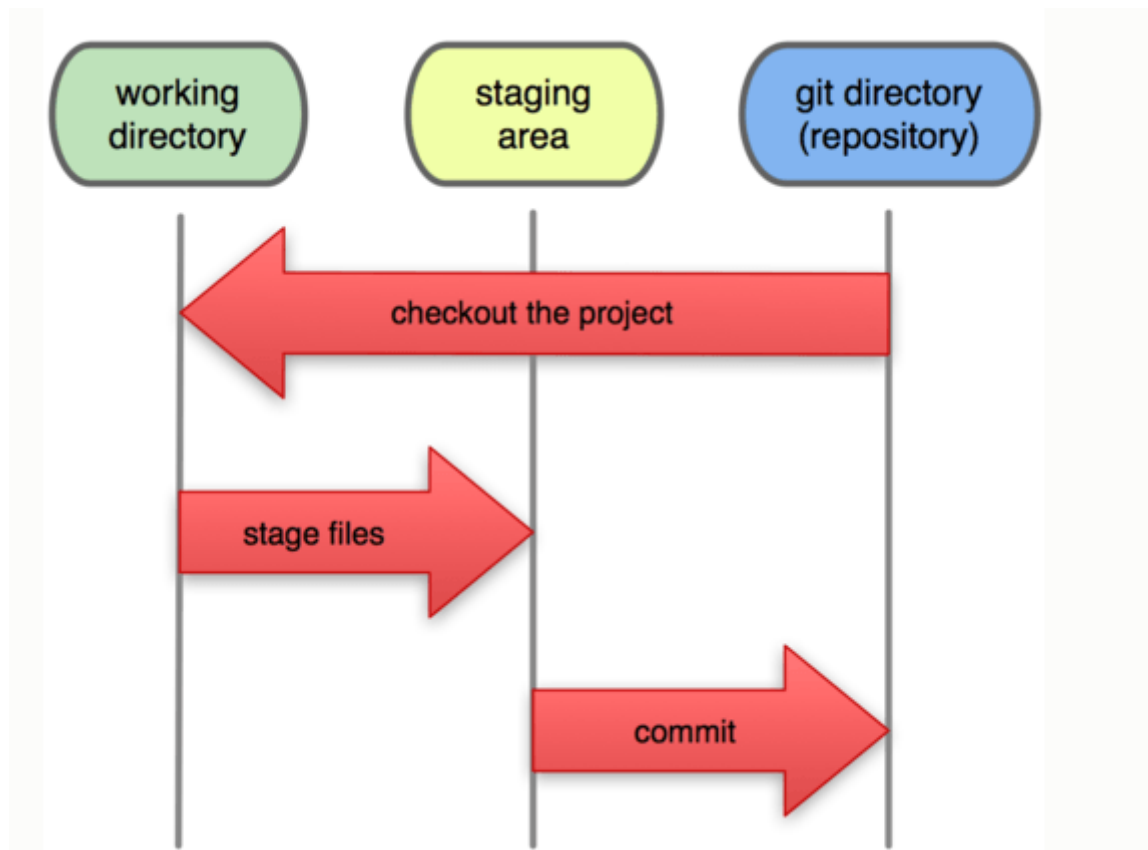
- **多数操作是添加数据**

常用的 Git 操作大多仅仅是把数据添加到数据库。Git 里，一旦提交快照之后就完全不用担心丢失数据，这种高可靠性令我们的开发工作安心不少，尽管去做各种试验性的尝试好了，再怎样也不会弄丢数据。

- **文件的三种状态**

接下来要讲的概念非常重要。对于任何一个文件，在 Git 内都只有三种状态：已提交 (committed)，已修改 (modified) 和已暂存 (staged)。已提交表示该文件已经被安全地保存在本地数据库中了；已修改表示修改了某个文件，但还没有提交保存；已暂存表示把已修改的文件放在下次提交时要保存的清单中。

由此我们看到 Git 管理项目时，文件流转的三个工作区域：Git 的工作目录，暂存区域，以及本地仓库。



基本的 Git 工作流程如下：

- 在工作目录中修改某些文件。
- 对修改后的文件进行快照，然后保存到暂存区域。git add
- 提交更新，将保存在暂存区域的文件快照永久转储到 Git 目录中。git commit

## 安装 git

这里提供下不同平台的 git 安装包下载地址,安装相信大家可以轻松搞定。

- Mac: <https://sourceforge.net/projects/git-osx-installer/>
- Windows: <https://git-for-windows.github.io/>
- Linux: apt-get install git