

引导语

同学，你好！我是徐国梁，欢迎你来到TD技能学院，这里是我的Git快速入门课。很高兴能和你一起学习版本控制技能。

作为开发者，代码版本控制是必须具备的一项技能。在开发工作中，代码版本控制技术不但可以更好地维护开发过程，也是协作编程的重要基础。

这一节课，我们将一起来学习以下这三个知识点：

- 版本控制的基本概念和意义
- Git版本控制系统的基本特征
- Git的安装与基础设置

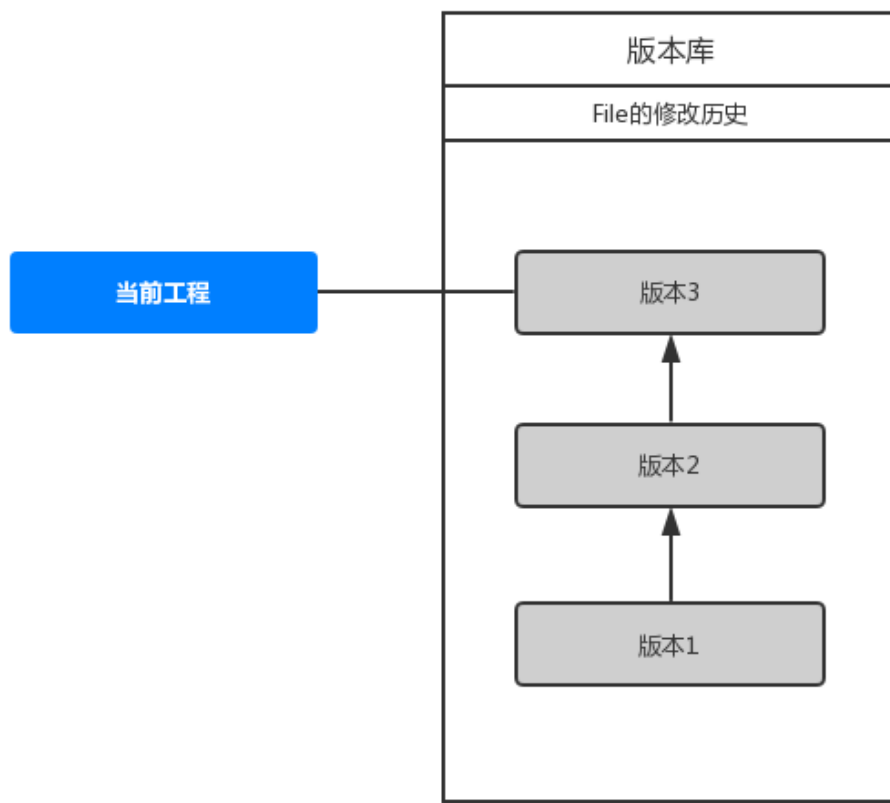
知识讲解

版本控制

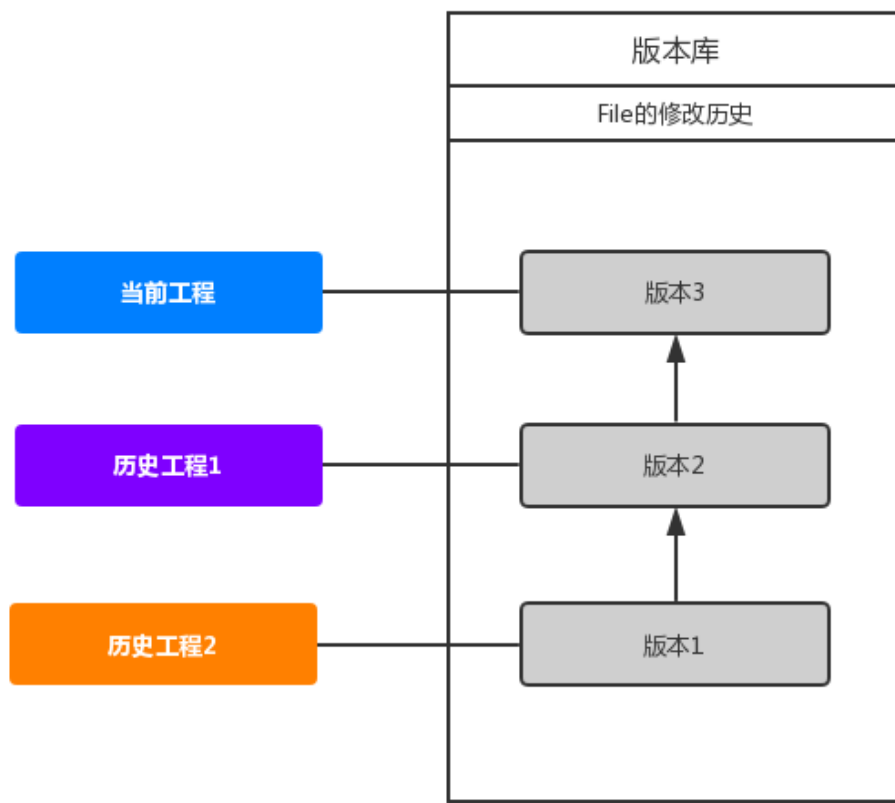
在正式开始介绍Git之前，我们需要先对版本控制有一个基本的概念。

版本控制，顾名思义，指的就是对工程文件的修改进行追踪记录的一种技术。所有对工程文件的修改将以版本为单位被记录，最终的结果由所有版本按时间顺序依次累加而得到。

如下面这张图所示



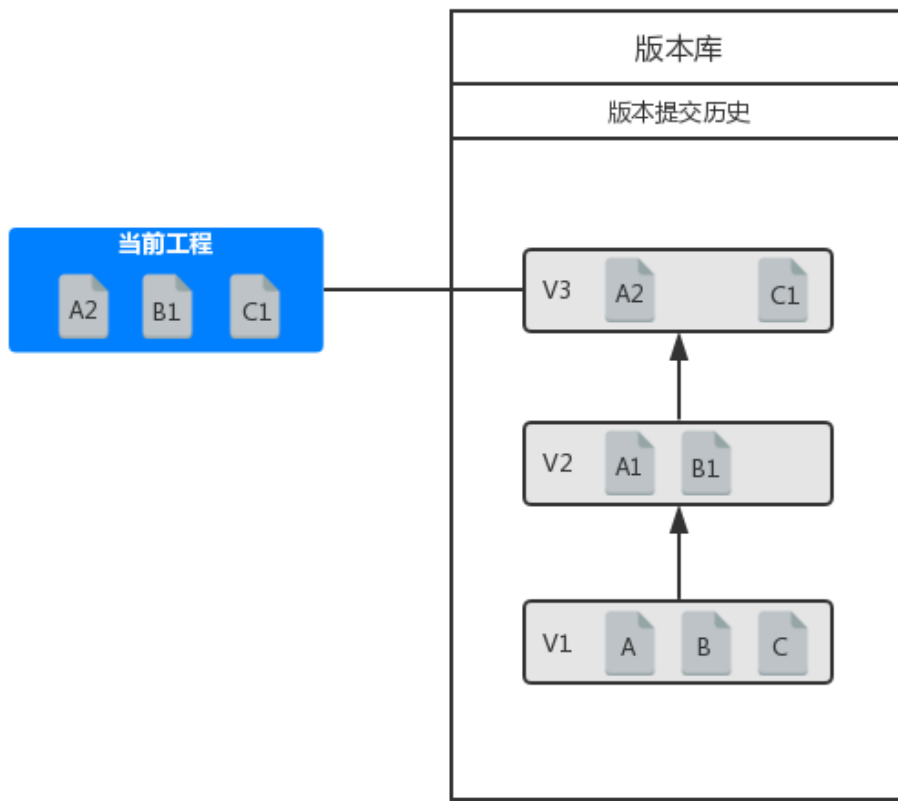
当前文件即是由所有的版本的修改累积而得到的。并且我们可以根据修改历史追溯到任何一个阶段，并看到当时项目的所有文件。



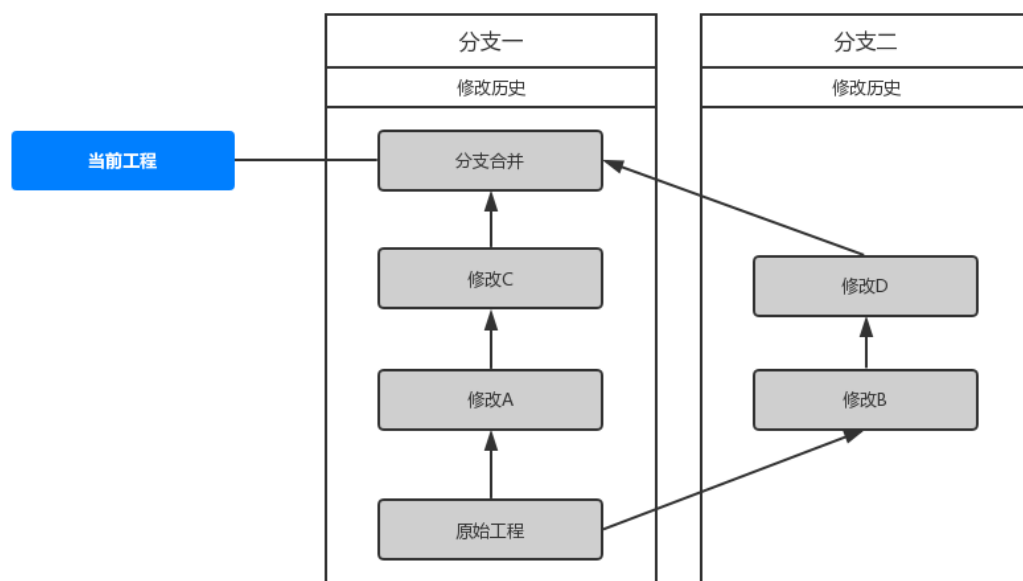
从前面两张图我们不难理解版本控制工具对于文件历史的追溯方式。在实际的开发中，绝大多数情况下，项目中都包含了多个文件。

因此版本控制工具对工程文件的版本记录大多是针对整个工程全局进行记录的，而非只是针对单个文件。

开发者每次提交修改的版本都可以包含对多个文件的修改，最终的工程由所有文件的修改历史共同累积的结果决定。如图



除了对一个线性的修改历史进行追踪外，版本控制系统一般都会支持多个分支共同修改，并在需要结果的时候进行合并。



这样一来，借助版本控制工具，我们不但可以很好地维护项目开发的过程，还为多人协作开发提供了基础。

在当今的软件工程领域，版本控制系统已经成为每一个正式项目依赖的基础服务。版本控制技术也已成为每一位工程师必须掌握的技巧。

Git的基本特性

下面我们就来认识一下Git这个号称当下最受欢迎的版本控制系统。一起探讨一下为何要选择Git，它有哪些过人的特性。

Git是目前世界上最先进的**分布式**版本控制系统，由Linux之父Linus Torvalds大神创造。

拥有最多的用户数量并管理着数量庞大的实际软件项目，当今全球最大的代码托管服务几乎全都是基于Git开发的。

作为版本控制系统，Git并不是最早被发明出来的。但是作为公认最强大的版本控制系统，git具备了很多强大的特性。

为了适应当今各式各样软件工程项目的需求，Git提供了几个强大的特性：

- 绝佳的性能
- 高度易用，以及非常易于理解的简单设计
- 对非线性开发模式的强力支持
- 完全分布式
- 精确到每个字符的记录完整性

首先是**性能**的优势：

Git具备同类产品最佳的运行速度，这主要依赖于Git在算法设计上选择了直接记录快照信息而非传统的差异比较的方式。

对于Git来说，文件的版本历史被理解为一个快照流，这样的设计免去了每次记录版本和取出版本时对比文件差异的操作。

与完整地保存每一个版本全部内容的方式不同，Git在存储修改历史的时候会将每一次修改的内容独立求出，并使用deflate算法对其进行压缩存储。

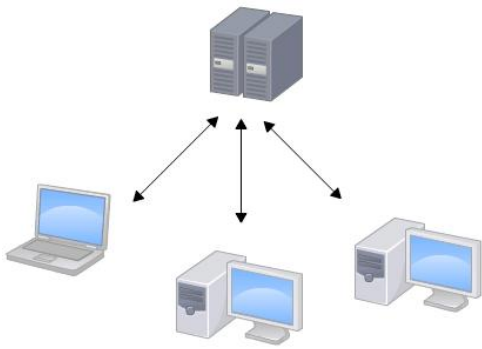
这样的设计使得git在内容存储及历史访问上都具备相当优秀的性能。

除了速度非常快之外，Git的结构也非常**易于理解**。这一点，在后续的课程中你将很快体验到。

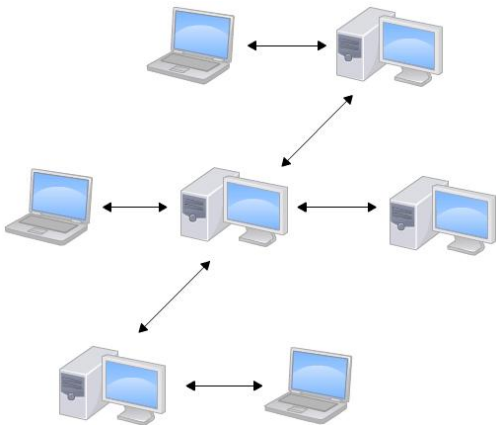
Git强大的**分支**能力可以支持成千上万个并行开发的分支。

最后，也是最重要的，Git是一种分布式版本控制系统，这意味着Git具备了“分布式”的特性。这是一种时髦的“去中心化”理念，解决了集中式版本控制严重依赖于服务器的问题，不会受到网络、服务器稳定性等不确定因素的影响。

这两张图对比了集中式版本控制系统和分布式版本控制系统在访问关系上的区别。



集中式



分布式

不难看出，相比之下分布式系统对中心服务器没有依赖，互相之间的依赖关系也更加的自由。

这种去中心化的理念使得每一个开发者都持有一个完整的版本仓库，在服务器宕机或网络受限的情况下，任何一个开发中的版本仓库都可以独立工作或以其为源重新建立代码分享中心。

这些特性使得它完全有能力高效管理类似 Linux 内核这样的超大规模项目。

Git的安装与基础设置

下面我们就来了解一下如何在我们的开发环境下安装配置Git。

这里我们以Windows系统为例。你可以直接在Git官网下载到最新的版本。访问这个网址，下载过程会自动进行。

<http://git-scm.com/download/win>

双击运行安装软件即可进行安装。在安装的过程中，你需要注意勾选把Git的运行路径添加到Path环境变量下。这样，在你安装完成之后，我们就能在开始菜单和右键菜单中看到Git自带的命令程序GitBash和交互式界面的版本Git GUI。

在初次使用Git之前，你需要做一些简单的配置，来让Git知道你是谁，以及你的偏好。每台计算机上只需要配置一次，之后你可以在任何时候再次通过运行命令来修改它们。

打开Git Bash，这是一个Git自带的命令行工具，它提供了很多相似于Linux下Bash命令行的命令。如果你熟悉Linux，使用起来就会很方便。

第一步，我们需要设置自己的用户信息。这是个必须的操作，因为每一次对版本进行操作的时候，Git都需要知道这是谁干的。

在Git Bash中写下这样两句代码

```
1 git config --global user.name "Aaron Xu"
2 git config --global user.email aaron@cineuse.com
```

就可以完成用户基本信息的设置。记得在你自己操作的时候把名字和邮箱改成自己的。

想要检查当前的配置信息，你可以直接输入 *git config <key>* 来检查对于key的配置。比如：

```
1 $ git config user.name
2 Aaron Xu
```

课程总结

好的，有关版本控制的基本概念以及Git的安装与配置就讲到这里了，最后让我们来一起回顾一下这节课讲到的知识点：

1. “版本控制”是一种追踪记录对工程文件的修改历史的技术，是开发项目可维护及可协作的基础
2. Git是一种分布式版本控制工具

3. Git安装完成后需要进行用户信息的配置

思考题

最后，请大家课后思考一下：相比于中心化的方式，分布式系统在协作上还有哪些优势？