**“Linux 生物信息技术基础”课程小组总结报告**

组：G12 次：8 组长：马小凯 执笔：郑弘熙

1. **时间**

2023年 4 月 23 日

1. **方式**

线上腾讯会议

1. **主题**

Git与GitHub介绍和使用

1. **内容**

**一、版本控制系统（VCSs）**

什么是版本控制？版本控制是一种记录一个或若干文件内容变化，以便将来查阅特定版本修订情况的系统。举个简单的例子：版本控制就是你写了一个文件，过了几天后，你对其进行了修改，但是你希望把原始的版本保留，同时保留修改后的，这其实就是一种版本控制。

一个具体的例子：想象一下你正在对某个重要的图片进行调整，例如你想看看只调饱和度的效果，但是你还希望看看只调整对比度的效果，这个时候，你就需要保存这一幅图片没修改、只调整饱和度和只调整对比度这三个修订版本，但是当问题变得复杂，你需要看看多种处理组合对图片的影响，并且对每种操作及操作后的图片进行记录保存，这其实就是一个版本控制的过程。

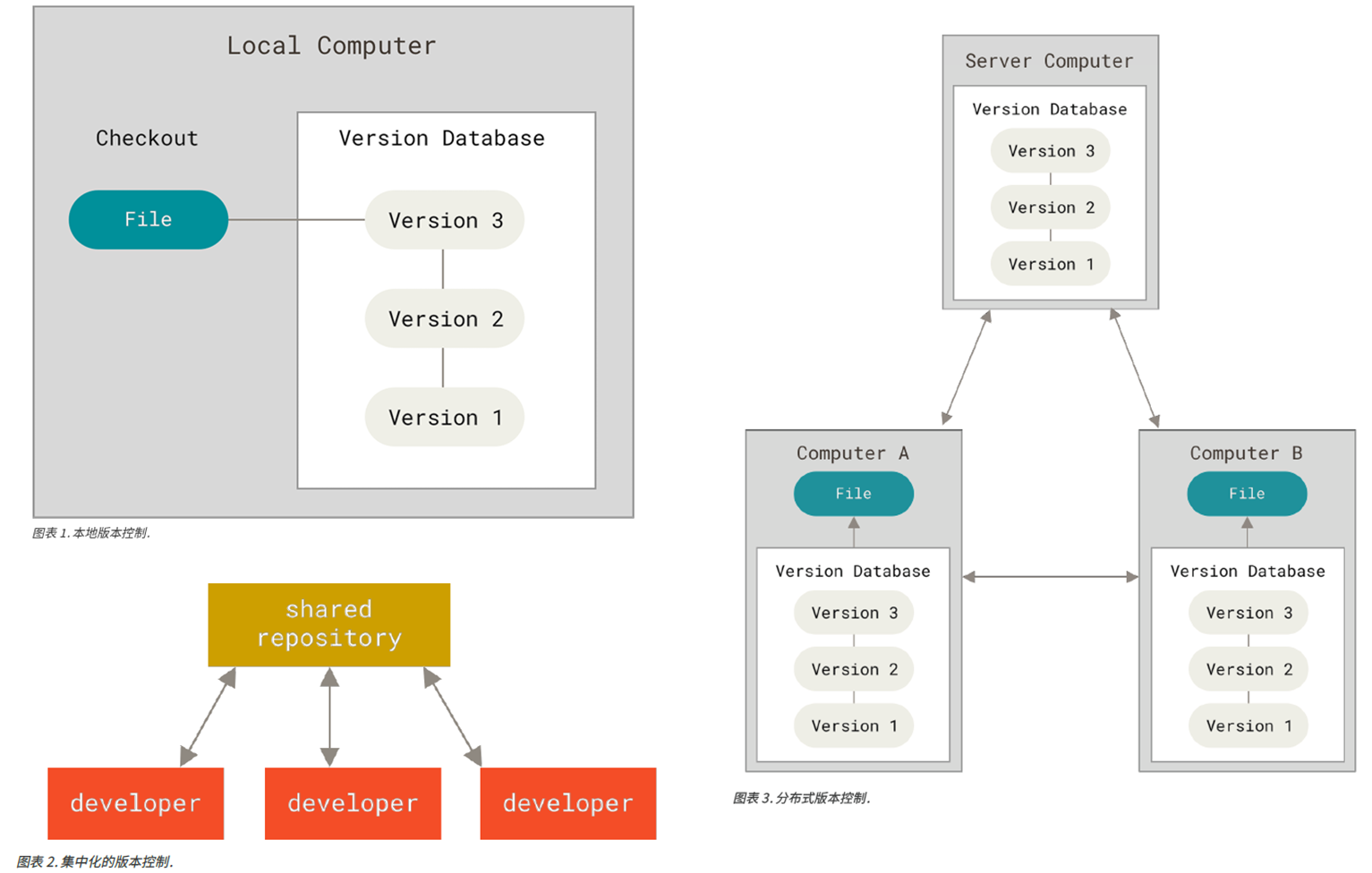
一般而言，很多人习惯用复制整个项目目录的方式来保存不同的版本，或许还会改名加上备份时间以示区别。这么做唯一的好处就是简单，但是特别容易犯错。因此，就有人开发了各种版本控制系统（Version Control Systems，VCSs）。

最开始的版本控制系统就是**本地端的版本控制系统**，其中最流行的一种叫做RCS，现今许多计算机系统上都还看得到它的踪影。RCS的工作原理是在硬盘上保存补丁集（补丁是指文件修订前后的变化）；通过应用所有的补丁，可以重新计算出各个版本的文件内容。

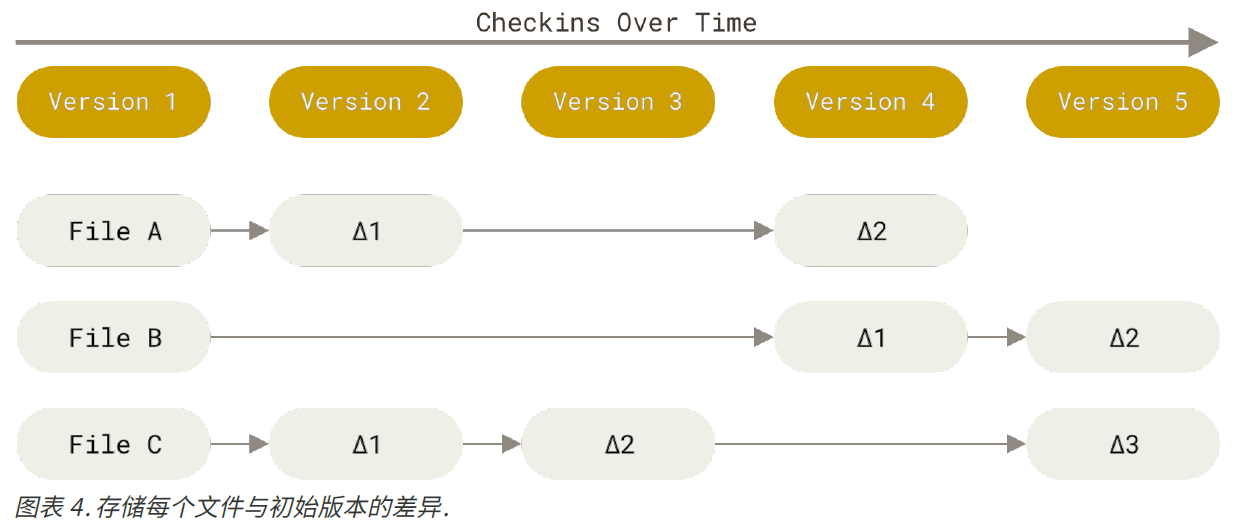
但是问题也随之而来，如何让在不同系统上的开发者协同工作？于是，**集中化的版本控制系统（Centralized Version Control Systems，简称 CVCS）**应运而生。这类系统，诸如CVS、Subversion以及Perforce等，都有一个单一的集中管理的服务器，保存所有文件的修订版本，而协同工作的人们都通过客户连到这台服务器，取出最新的文件或者提交更新。

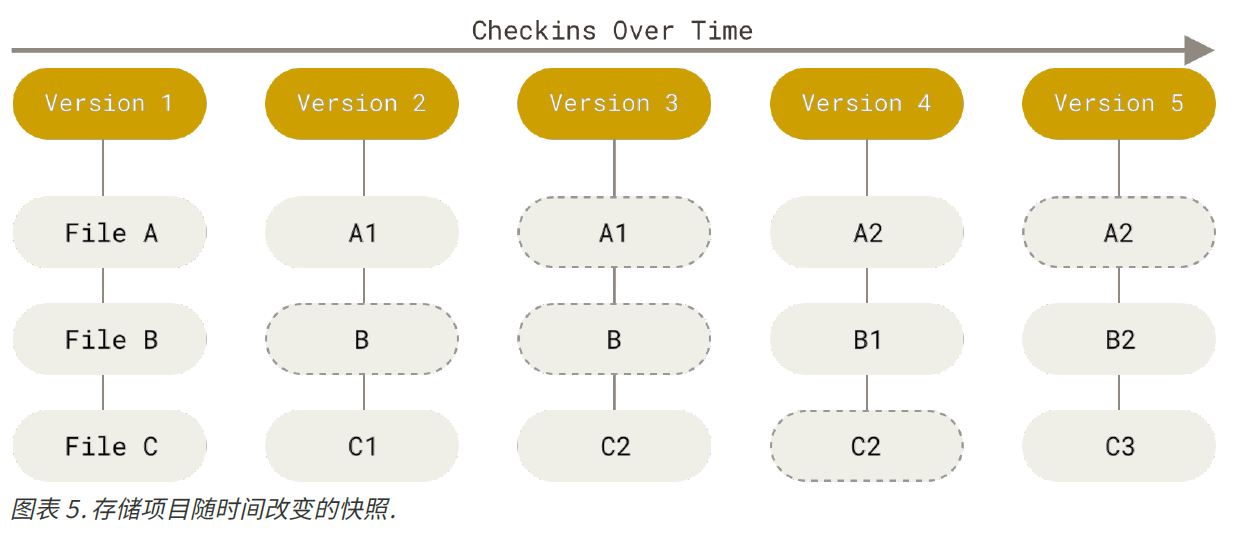
但很显然的一点是当中央服务器出现故障。那么在故障的这段时间内，谁都无法上传更新，无法协同工作。如果中心数据库所在的磁盘发生损坏，又没有做恰当备份，毫无疑问你将丢失所有数据——包括项目的整个变更历史，只剩下人们在各自机器上保留的单独快照。

因此**分布式版本控制系统（Distributed Version Control System，简称 DVCS）**面世了。在这类系统中，像Git、Mercurial、Bazaar以及 Darcs等，客户端并不只提取最新版本的文件快照，而是把代码仓库完整地镜像下来，包括完整的历史记录。这么一来，任何一处协同工作用的服务器发生故障，事后都可以用任何一个镜像出来的本地仓库恢复。因为每一次的克隆操作，实际上都是一次对代码仓库的完整备份。

**二、Git概述**

在对版本控制系统有了一定的认识后，包括为什么需要，都有哪几种，是怎么发展的，都有什么利弊。我们接下来就可以开始介绍Git了。Git它不仅仅是个版本控制系统，它也是个内容管理系统（CMS），工作管理系统等。它是Linus Torvalds为了帮助管理Linux内核开发而开发的一个开放源码的版本控制软件。作为一个开源的分布式版本控制系统，它被用于敏捷高效地处理任何或小或大的项目。与常用的版本控制工具 CVS, Subversion等不同，它采用的分布式版本库的方式，不必服务器端软件支持，使源代码的发布和交流极其方便。

**Git和其它版本控制系统（包括Subversion及其他近似工具）的主要差别在于Git对待数据的方式。**从概念上来说，其它大部分系统以文件变更列表的方式存储信息，这类系统（CVS、Subversion、Perforce、Bazaar等等）将它们存储的信息看作是一组基本文件和每个文件随时间逐步累积的差异（它们通常称作基于差异（delta-based）的版本控制）。

Git不按照以上方式对待或保存数据。反之，Git更像是把数据看作是对小型文件系统的一系列快照。在Git中，每当你提交更新或保存项目状态时，它基本上就会对当时的全部文件创建一个快照并保存这个快照的索引。为了效率，如果文件没有修改，Git不再重新存储该文件，而是只保留一个链接指向之前存储的文件。**Git对待数据更像是一个快照流。**

**三、Git的安装**

Git目前支持Linux/Unix、Solaris、Mac和Windows平台上运行。Git各平台安装包下载地址为：<http://git-scm.com/downloads>。具体可以参照菜鸟教程：https://www.runoob.com/git/git-install-setup.html

**（1）Linux平台上安装**

Git的工作需要调用curl，zlib，openssl，expat，libiconv等库的代码，所以需要先安装这些依赖工具。

各Linux系统可以使用其安装包管理工具（apt-get、yum等）进行安装，或者在官网下载源码包进行安装。

**（2）Windows上安装**

官方版本可以在Git官方网站下载。打开<https://git-scm.com/download/win>，下载会自动开始。要注意这是一个名为Git for Windows的项目（也叫做msysGit），和Git是分别独立的项目，更多信息可以访问<http://msysgit.github.io/>。

也可以在如下网站下载Git for Windows：<https://gitforwindows.org/>。完成安装之后，就可以使用命令行的Git工具（自带ssh客户端），另外还有一个图形界面的Git项目管理工具。在开始菜单里找到"Git"->"Git Bash"，会弹出Git命令窗口，你可以在该窗口进行Git操作。

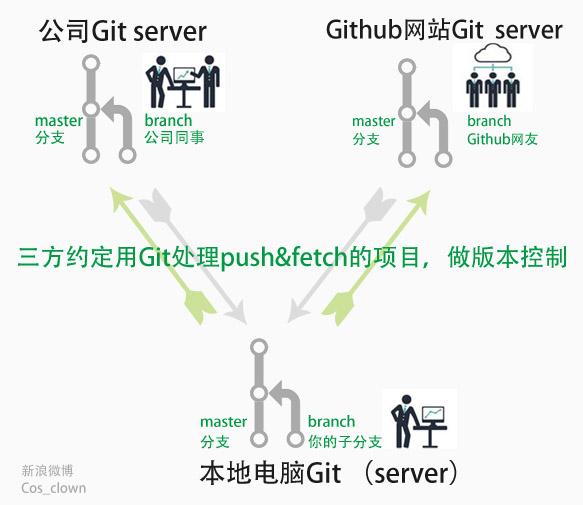
另一个简单的方法是安装GitHub Desktop。该安装程序包含图形化和命令行版本的Git。它也能支持Powershell，提供了稳定的凭证缓存和健全的换行设置。可以在GitHub for Windows网站下载，网址为：<https://desktop.github.com/>。

**（3）macOS上安装**

一种是安装Xcode Command Line Tools。Mavericks（10.9）或更高版本的系统中，只需要在Terminal里尝试首次运行git命令即可。如果没有安装过命令行开发者工具，将会提示你安装。

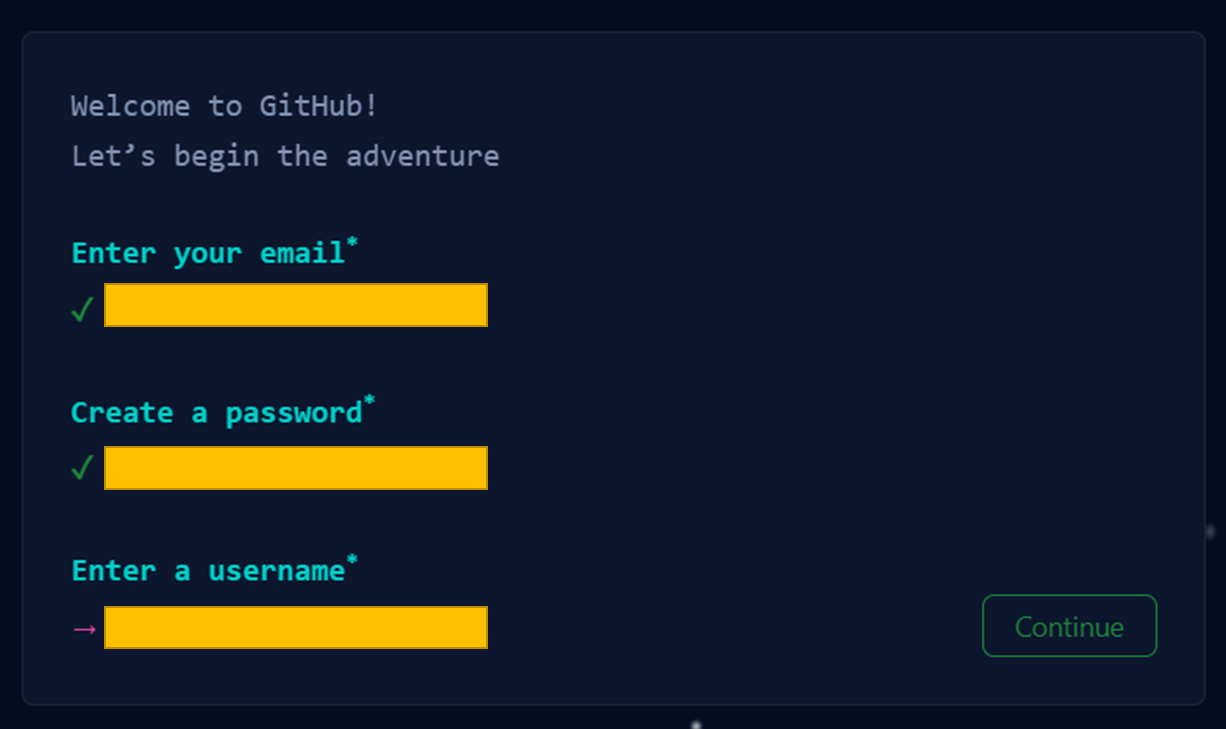
在Mac平台上安装Git另一种比较容易的方法就是使用图形化的Git安装工具，下载地址为：<http://sourceforge.net/projects/git-osx-installer/>

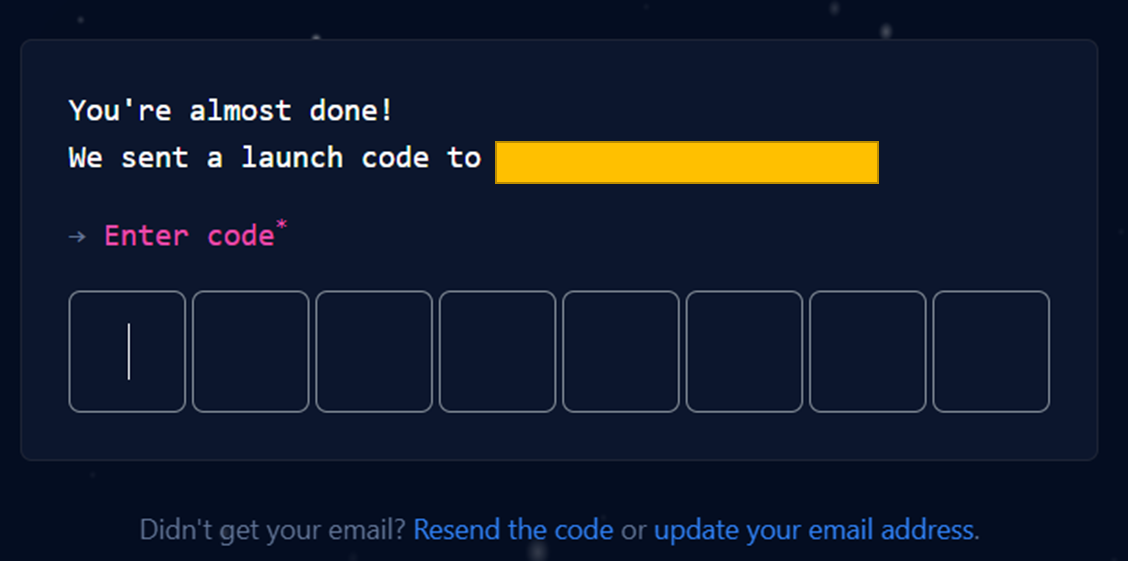
**四、Git与GitHub**

GitHub是一个面向开源及私有软件项目的托管平台，因为只支持Git作为唯一的版本库格式进行托管，故名GitHub。简单地说，Git是上面提到的版本控制工具中的一种，而GitHub则是一个用Git做版本控制的项目托管平台。如图可以比较好地理解二者关系。

**五、GitHub的注册**

前提：似乎需要“科学上网”/“翻墙”

登录<https://github.com/>，选择Sign up，依次输入邮箱（可以收到邮件）、GitHub密码，用户名，如下图所示。完成后，会询问你是否接受产品更新通知邮件，并且进行人机验证。完成后，点击最下方的create account进入下一步。

 需要你进行邮箱验证，先前填写的邮箱会收到验证码，填写进去即可。之后会有一个使用需求问卷，根据个人情况填写即可。并且会有一个询问你是否需要pro版本的GitHub，一般而言选择免费的版本即可。

完成这些步骤，你的GitHub账号就注册完成咯~

**最后的最后！推荐两个网站**

Git教程：<https://git-scm.com/book/zh/v2>

PDF：<https://github.com/progit/progit2-zh/releases/download/2.1.62/progit.pdf>

菜鸟教程：<https://www.runoob.com/git/git-tutorial.html>