狂神聊Git

使用git之前，我们需要先明白一个概念，版本控制！

一、版本控制

1. 什么是版本控制 版本迭代，新的版本！版本管理器

版本控制 (Revision control) 是一种在开发过程中用于管理我们对文件、目录或工程等内容的修改，方便查看更改历史记录，备份以便恢复以前的版本的软件工程技术

实现跨区域多人协同开发

追踪和记录一个或者多个文件的历史记录

组织和保护你的源代码和文档

统计工作量

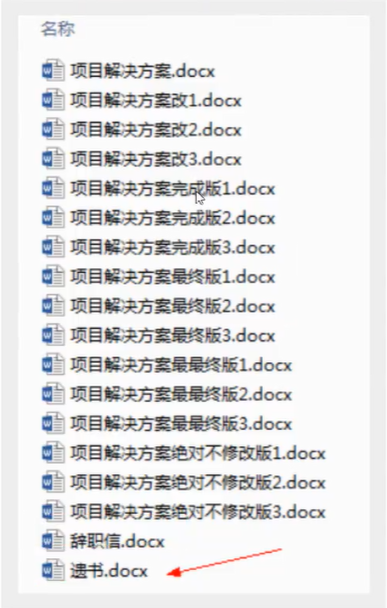
并行开发、提高开发效率

减轻开发人员的负担，节省时间，同时降低人为错误

简单说就是用于管理多人协同开发项目的技术

没有进行版本控制或者版本控制本身缺乏正确的流程管理，在软件开发过程中会引入很多问题，如软件代码的一致性、软件内容的冗余、软件过程的事物性、软件发开过程中的并发性、软件源代码的安全性，以及软件的整合等问题

无论是工作还是学习，或者是自己做笔记，都经历过这样一个阶段！我们就迫切需要一个版本控制工具！



多人开发就必须要使用版本控制，代价会比较大

1. 常见的版本控制工具

我们学习的东西，一定是当下最流行的！

主流的版本控制器有如下这些：

Git

SVN ( Subversion )

CVS ( Concurrent Versions System )

VSS ( Micorosoft Visual SourceSafe )

TFS ( Team Foundation Server )

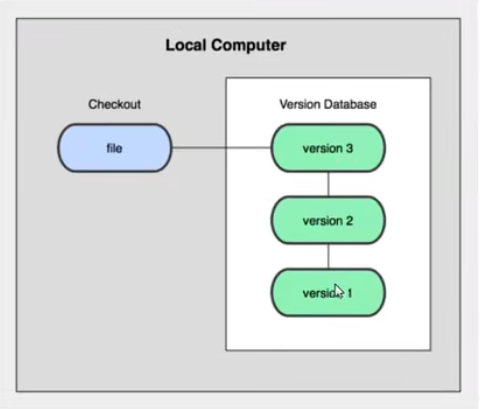
Visual Studio Online

版本控制产品非常的多 ( Peoforce、Rational ClearCase、RCS ( GNU Revision Control System )、Seren Dimention、SVK、BitKeeper、Monotone、Bazaar、Mercurial、SourceGear Vault )，现在影响力最大且最广泛的是Git和SVN

1. 版本控制分类

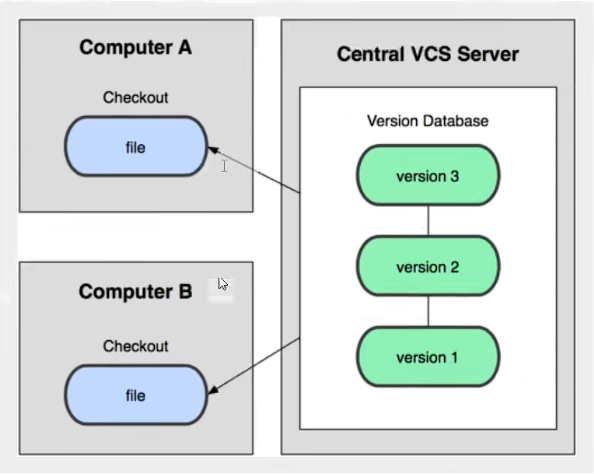
**①本地版本控制**

记录文件每次的更新，可以对每个版本做一个快照，或是记录补丁文件，适合个人用，如RCS



**②集中版本控制**

所有的版本数据都保存在服务器上，协同开发者从服务器上同步更新或上传自己的修改



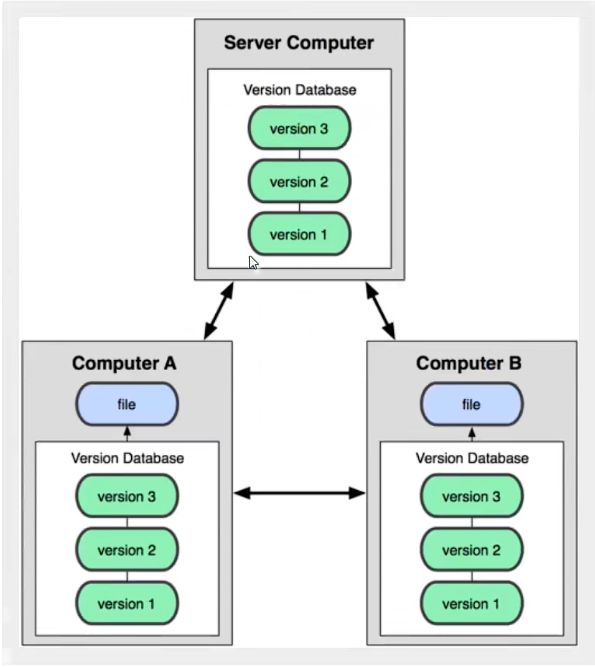
所有的版本数据都存在服务器上，用户的本地只有自己以前所同步的版本，如果不连网的话，用户就看不到历史版本，也无法切换版本验证问题，或在不同分支工作。而且，所有数据都保存在单一的服务器上，有很大的风险这个服务器会损坏，这样就会丢失所有的数据，当然可以定期备份。代表产品:SVN、CVS、VSS

**③分布式版本控制 Git**

每个人都拥有全部的代码！安全隐患！

所有版本信息仓库全部同步到本地的每个用户，这样就可以在本地查看所有版本历史，可以离线在本地提交，只需在连网时push到相应的服务器或其他用户那里。由于每个用户那里保存的都是所有的版本数据，只要有一个用户的设备没有问题就可以恢复所有的数据，但这增加了本地存储空间的占用。

不会因为服务器损坏或者网络问题，造成不能工作的情况！



1. Git与SVN最主要的区别

SVN是集中式版本控制系统，版本库是集中放在中央服务器的，而工作的时候，用的都是自己的电脑，所以首先要从中央服务器得到最新的版本，然后工作，完成工作后，需要把自己做完的活推送到中央服务器。集中式版本控制系统是必须联网才能工作，对网络带宽要求较高。



Git是分布式版本控制系统，没有中央服务器，每个人的电脑就是一个完整的版本库，工作的时候不需要联网了，因为版本都在自己电脑上。协同的方法是这样的︰比如说自己在电脑上改了文件A，其他人也在电脑上改了文件A，这时，你们两之间只需把各自的修改推送给对方，就可以互相看到对方的修改了。Git可以直接看到更新了哪些代码和文件!

Git是目前世界上最先进的分布式版本控制系统。

二、聊聊Git历史

同生活中的许多伟大事物一样，Git诞生于一个极富纷争大举创新的年代。

Linux内核开源项目有着为数众广的参与者。绝大多数的Linux内核维护工作都花在了提交补丁和保存归档的繁琐事务上(1991-2002年间)。到2002年，整个项目组开始启用一个专有的分布式版本控制系统BitKeeper来管理和维护代码。

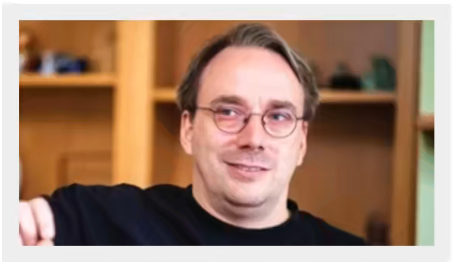
Linux社区中存在很多的大佬!破解研究BitKeeper !

到了2005年，开发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Linux内核社区免费使用。

BitKeeper的权力。这就迫使Linux开源社区(特别是Linux的缔造者Linus Torvalds)基于使用BitKeeper时的经验教训，开发出自己的版本系统。（2周左右！）也就是后来的Git !

Git是目前世界上最先进的分布式版本控制系统。

Git是免费、开源的，最初Git是为辅助Linux内核开发的，来替代 BitKeeper !



Linux和Git之父李纳斯·托沃兹（Linus Benedic Torvalds ) 1969、芬兰

1. Git环境配置
2. 软件下载

打开[git官网] <https://git-scm.com/> ，下载git对应操作系统的版本。

所有东西下载慢的话就可以去找镜像!

官网下载太慢，我们可以使用淘宝镜像下载: <http://npm.taobao.org/mirrors/git-for-windows/>

下载对应的版本即可安装!

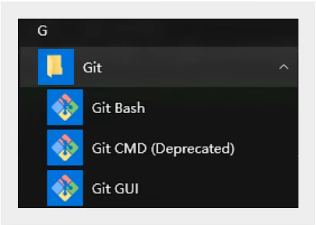
安装:无脑下一步即可! 安装完毕就可以使用了！

1. 先卸载

直接反安装即可、然后清理环境变量

1. 启动Git

安装成功后在开始菜单中会有Git项，菜单下有3个程序∶任意文件夹下右键也可以看到对应的程序!



**Git Bash** : Unix与Linux风格的命令行，使用最多，推荐最多

**Git CMD** : Windows风格的命令行

**Git GUI**:图形界面的Git，不建议初学者使用，尽量先熟悉常用命令

1. 基本的Linux命令学习

平时一定要多使用这些基础的命令！

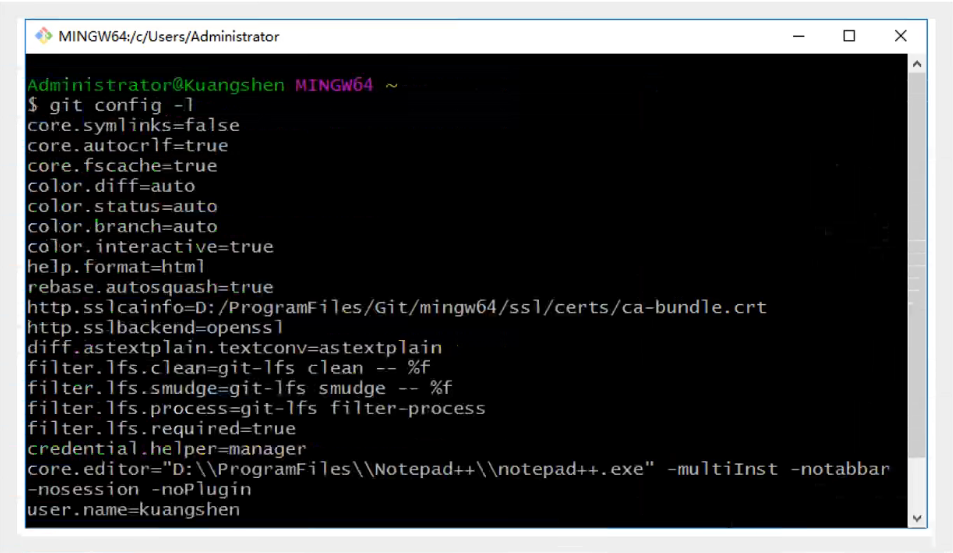
1. cd: 改变目录
2. cd .. : 回退到上一个目录，直接cd进入默认目录
3. pwd: 显示当前所在的目录路径
4. ls(ll): 都是列出当前目录中的所有文件，只不过K(两个II)列出的内容更为详细。
5. touch: 新建一个文件如touch index.js就会在当前目录下新建一个index.js文件。
6. rm: 删除一个文件, rm index.js 就会把index.js文件删除。
7. mkdir: 新建一个目录,就是新建一个文件夹。
8. rm -r: 删除一个文件夹, rm -r src删除src目录

rm -rf / 切勿在Linux中尝试！删除电脑中全部文件！

1. mv 移动文件, mv index.html src index.html是我们要移动的文件, src是目标文件夹,当然,这样写,必须保证文件和目标文件夹在同一目录下
2. reset: 重新初始化终端/清屏。
3. clear: 清屏
4. history: 查看命令历史
5. help: 帮助
6. exit: 退出
7. #表示注解
8. Git配置

所有的配置文件，其实都保存在本地！

查看配置 git config -l



查看不同级别的配置文件

#查看系统config

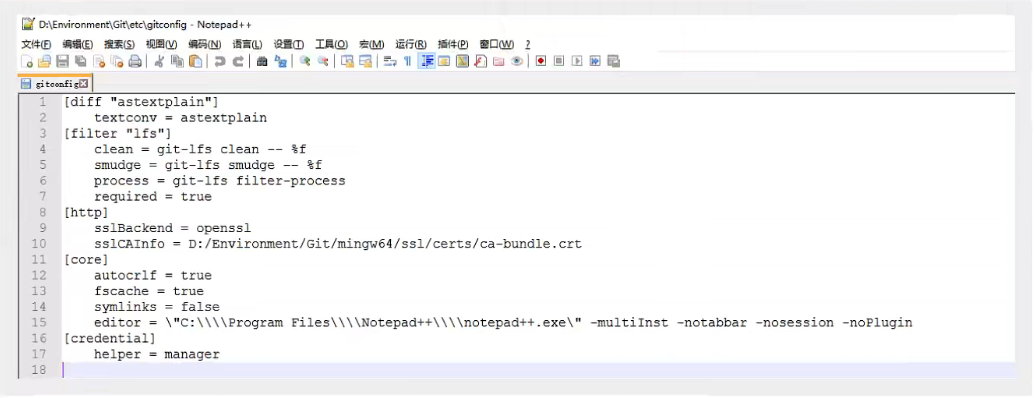
Git config --system --list

#查看当前用户（global）配置

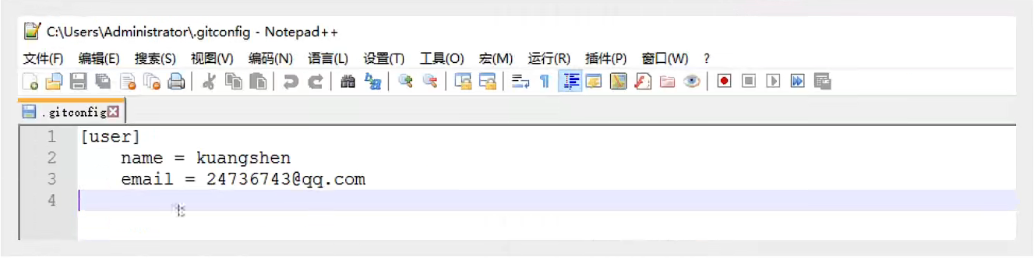
Git config --global --list

**Git相关的配置文件：**

1 ) Git\etc\gitconfig : Git安装目录下的gitconfig --system系统级



2 ) C:\Users\Administrator\gitconfig只适用于当前登录用户的配置 --global全局



这里可以直接编辑配置文件，通过命令设置后会响应到这里。

1. 设置用户名与邮箱 (用户标识，必要)

当你安装Git后首先要做的事情是设置你的用户名称和e-mail地址。这是非常重要的，因为每次Git提交都会使用该信息。它被永远的嵌入到了你的提交中∶

git config --global user.name "kuangshen" #名称

git config --global user.email 24736743@qq.com #邮箱

只需要做一次这个设置，如果你传递了--global选项，因为Git将总是会使用该信息来处理你在系统中所做的一切操作。如果你希望在一个特定的项目中使用不同的名称或e-mail地址，你可以在该项目中运行该命令而不要--global选项。总之--global为全局配置，不加为某个项目的特定配置。

