1.什么是Git（傻瓜内容跟踪器）

     Git是一款免费、开源的分布式版本控制系统，用于敏捷高效地处理任何或小或大的项目

Git 是 Linus Torvalds （Linus在1991年创建了开源的Linux）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。

2.Git发展的历史：

     1.在2002年以前，世界各地的志愿者把源代码文件通过diff的方式发给Linus，然后由Linus本人通过手工方式合并代码！（因为Linus坚定地反对CVS【CVS（Concurrent Version System）是一个C/S系统、一个常用的代码[版本控制软件](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=7971489" \t "http://baike.sogou.com/_blank)。CVS[版本控制系统](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=576892" \t "http://baike.sogou.com/_blank)是一种GNU[软件包](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=48991" \t "http://baike.sogou.com/_blank)，主要在开源[软件管理](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=2203984" \t "http://baike.sogou.com/_blank)中使用，在多人[开发环境](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=73226299" \t "http://baike.sogou.com/_blank)下对源码进行维护。】和SVN【SVN是Subversion的简称，是一个[开放源代码](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=269184" \t "http://baike.sogou.com/_blank)的[版本控制系统](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=576892" \t "http://baike.sogou.com/_blank)，相较于RCS、CVS，它采用了分支管理系统，它的设计目标就是取代CVS。互联网上很多[版本控制](http://baike.sogou.com/lemma/ShowInnerLink.htm?lemmaId=576892" \t "http://baike.sogou.com/_blank)服务已从CVS迁移到Subversion。说得简单一点SVN就是用于多个人共同开发同一个项目，共用资源的目的。】，这些集中式的版本控制系统不但速度慢，而且必须联网才能使用）

     2.到了2002年，Linux系统已经发展了十年了，代码库之大让Linus很难继续通过手工方式管理了，社区的弟兄们也对这种方式表达了强烈不满，于是Linus选择了一个商业的版本控制系统BitKeeper，BitKeeper的东家BitMover公司出于人道主义精神，授权Linux社区免费使用这个版本控制系统。

     3.安定团结的大好局面在2005年就被打破了，原因是Linux社区牛人聚集，不免沾染了一些梁山好汉的江湖习气。开发Samba的Andrew试图破解BitKeeper的协议（这么干的其实也不只他一个），被BitMover公司发现了（监控工作做得不错！），于是BitMover公司怒了，要收回Linux社区的免费使用权。

Linus可以向BitMover公司道个歉，保证以后严格管教弟兄们，嗯，这是不可能的。实际情况是这样的：

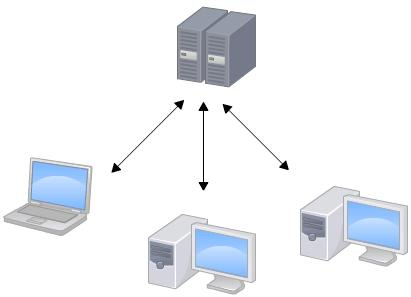
     4.Linus花了两周时间自己用C写了一个分布式版本控制系统，这就是Git！一个月之内，Linux系统的源码已经由Git管理了！牛是怎么定义的呢？大家可以体会一下。

     Git迅速成为最流行的分布式版本控制系统，尤其是2008年，GitHub网站上线了，它为开源项目免费提供Git存储，无数开源项目开始迁移至GitHub，包括jQuery，PHP，Ruby等等。

     历史就是这么偶然，如果不是当年BitMover公司威胁Linux社区，可能现在我们就没有免费而超级好用的Git了。

3.与SVN的对比

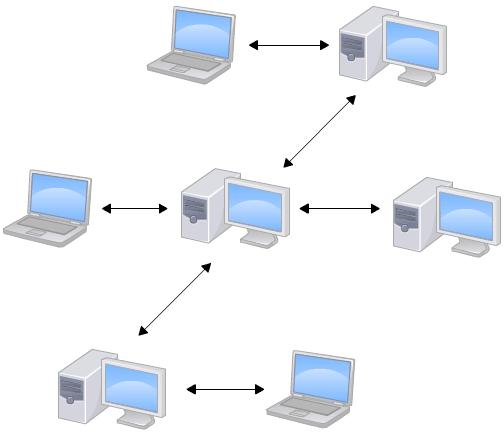
集中式版本控制系统（中央服务器）



缺点：

     1.每次提交代码都需要联网（没网，或者网速比较慢）

     2.每次提交都是直接提交到中央服务器，自己做项目不可能一气呵成，却没有自己的代码管理版本

分布式版本控制系统（每个人的电脑都有一个库）

 优点

1.不需要联网（相对的，比如最终代码的合并是需要联网的）

2.可以有一个自己的版本库，不污染其他成员的代码，也方便自己的管理（强大的分支管理）

4.Git安装

     Ubuntu 14.04安装:sudo apt-get install git

     Windows下载

     git config —global user.name “lsy"

     git config —global user.email “1635929054@qq.com"

创建版本库

     1. 新建目录 admin

     2. git init：将此目录变为git可以管理的仓库

          .git:跟踪管理版本的，不要随便修改里面的文件

文件操作

     git status     查看当前工作区状态

     git add index.php     添加文件到暂存区

     git commit -m “第一个文件” 提交文件到版本库，并做说明

     git diff index.php     比较index.php的前后版本不同地方

     git log     查看历史版本

          git log --pretty=oneline     只输出一行

     git reset —hard  HEAD^     回退到上一个版本

               git reset —hard HEAD~10 回退上10次

               git reset —hard  3C3981     回退到指定的版本

     git reflog     查看版本历史

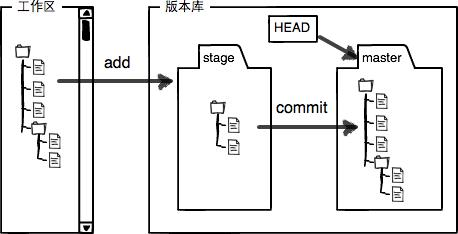
工作区和暂存区

     工作区：admin目录就是工作区

     版本库：.git

     暂存区：state

     head：指针

     master：当前分支

git add index.php     将文件添加到暂存区

git commit -m “修改"     将暂存区的文件合并到当前的分支

git跟踪管理的是修改，而非文件

vi index.php(修改index.php)

git add index.php(将index.php文件添加到暂存区)

vi index.php（再次修改index.php）

git commit -m “修改"

撤销修改：

git checkout — index.php     撤销index.php在工作去的修改

git reset HEAD index.php     将暂存区的文件撤销，重新放回工作区（但是修改的内容并不会删除，需要删除，还需要git checkout —index.php）

删除文件

git add index.php

git commit -m “添加index.php"

两种办法

     1.rm index.php     直接删除

     2.git rm index.php

          git commit -m “delete index.php"

撤销办法：

    删除文件，还没有提交：直接使用git checkout — index.php

    已经提交，找回删除文件（版本会退）：使用 git reset —hard HEAD^