Linus自己用C语言开发Git

版本控制工具具有的功能：

协同修改

数据备份

版本管理

SVN采用增量式管理，Git采用了文件系统快照方式

权限控制

历史记录

分制管理

集中式版本控制工具：SVN

简介：文件和版本信息存储在服务器上，开发者和服务器进行交互

缺点：服务器宕机信息会丢失，（单点故障）

分布式版本控制工具：Git

简介：本地可以进行版本控制，本地可以进行互传

优点：避免单点故障

Git的优势：

1. 大部分操作在本地完成，不需要联网
2. 完整性保证
3. 尽可能添加数据而不是删除或修改数据
4. 分支操作非常快捷流畅
5. 与Linux命令全面兼容

Git在本地的结构

工作区 git add 暂存区 git commit 本地库

Git和代码托管中心：维护远程库

局域网环境下:GitLab服务器

外网环境下

GitHub

本地初始化 git init 创建一个.git目录

.git目录中存放的事本地库相关的子目录和文件，不要删除和修改

设置签名 git config --global

git config --global

保存的位置 ~/.gitconfig文件

项目级别：仅在当前本地库范围有效

系统用户级别：登录当前操作系统的用户范围

项目级别优先于系统用户级别

Git命令

git status（查看当前状态）工作区和暂存区

git add文件名 （将工作区添加到暂存区的操作）

git rm --cached 文件名 （从暂存区删除的操作）

git commit 文件名 (从暂存区提交到本地库)

git commit -m “注释内容” 文件名

查看历史记录

git log

多屏显示控制方式：空格向下，b向上翻页，q退出

git log --pretty ==oneline (一行显示)

git log --oneline

git reflog 到某个版本需要移动多少步

基于索引值前进后退版本形式

git reset --hard(局部索引值) 建议使用

使用‘^’‘~’ 只能往后退

git reset --hard HEAD^ （一个^退后一步）

git reset --hard HEAD~N （表示后退N步）N>1

reset 参数: soft,仅仅在本地移动HEAD指针

mixed：在本地库移动HEAD指针，重置暂存区

hard：在本地库移动HEAD指针，重置暂存区，重置工作区

git reset --hard HEAD 刷新指针所在的位置

删除文件并找回

删除前，文件存在时的状态提交到了本地库

操作， git reset --hard 指针的位置

git diff(文件名）：将工作区的文件和暂存区进行比较

git diff(本地库中历史版本）（文件名）：将工作区中的文件和本地库历史记录比较

分支操作

git branch -v 查看所有的分支

git branch [分支名字]创建新的分支

git checkout [分支名字] 切换分支

合并分支：1，切分到接受修改的分支，执行 git checkout[被合并分支名]

2, 执行merge 命令 ，git merge[有新内容分支名]

冲突的解决：

1. 编辑文件，删除特殊符号
2. 把文件修改到满意的程度，保存退出
3. git add [文件名]
4. git commit - m ‘日志信息’（注意，此时commit一定不能带具体文件名）

哈希，

1. 不管输入数据的数据量有多大，输入同意哈希算法，得到的加密结果长度固定
2. 哈希算法确定，输入数据确定，输出数据能够保持不变
3. 哈希算法确定，输入数据幼斌华，输出数据一定有变化，而且通常变化很大
4. 哈希算法不可逆

哈希算法可以用来验证文件。

Svn 增量式版本控制

Git 快照流版本控制

GitHub

通过git上传远程库（GitHub）

创建GitHub别名git remote add (别名) +GitHub地址

查看别名 git remote -v

把本地库上传远程库 git push 别名 +指定分支

把远程库克隆到本地库 git clone + 远程库地址

克隆效果（clone）:1,完整的把远程库下载到本地，2，创建\*\*远程地址别名，3初始化本地库

拉取合并 pull = fetch +merge

只拉取git fetch [远程地址别名][远程分支名]

合并 git merge [远程地址别名][远程分支名]

解决冲突

1. 如果不是基于Git Hub 远程库的最新版所做的修改，不能推送，必须先拉取
2. 拉取下来后如果进入冲突状态，则按照‘分支冲突解决’操作解决即可

跨团队协作

参与者直接fork

本地修改，然后推送到远程

Pull request

SSH登录 登录当前目录的家目录

删除.ssh目录rm -rvf.ssh

ssh -keygen -t rsa -C GitHub账号

进入ssh目录查看文件列表 查看id\_rsa.pub 复制文件内容登录GitHub

点击用户头像 setting 选择SSH and GPGkeys 点击New SSH key 复制到key

新建远程库别名 使用SSH地址 git remote add +SSH地址