Git&Github

1. 版本控制工具应该具备的功能

* 协同修改

多人并行不悖的修改服务器的文件

* 数据备份

保存每一个提交过的历史状态

* 版本控制

保存文件系统的快照（？）

* 权限控制

团队内控制，团队外贡献的代码审核

* 历史记录

记录修改人，修改时间，内容

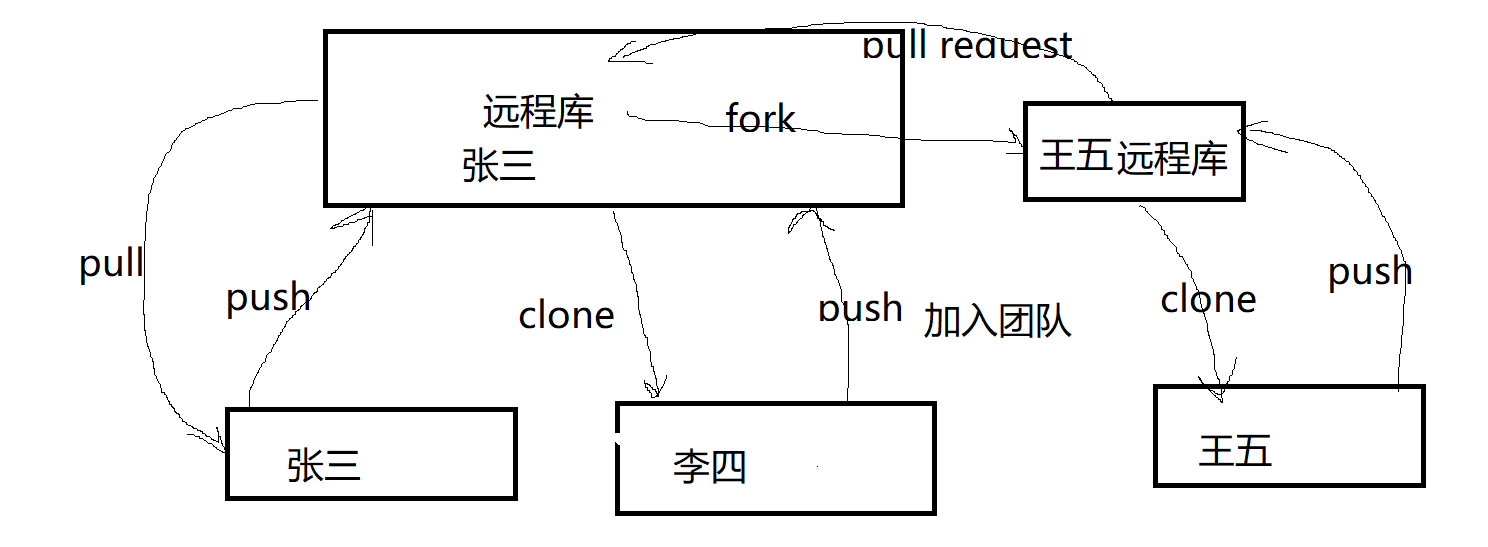
恢复到某一个历史状态

* 分支管理

多条生产线同时进行，提高效率

3.7本地库和远程库

* 团队内部协作
* 团队外部协作



4.Git命令行操作

4.1Git初始化

* 命令

git init

在某个文件夹下用此命令，创建一个空的仓库，还有一个.git

4.2设置签名

* 和登录github的用户名没有关系
* 命令

项目级别：仅在当前本地库的目录有效

git config

系统用户级别：登录当前操作系统的用户范围

git config –global

4.3 **git rm --cached** <文件名> 从暂存区删除

**git checkout** --<file> 撤销修改(?)

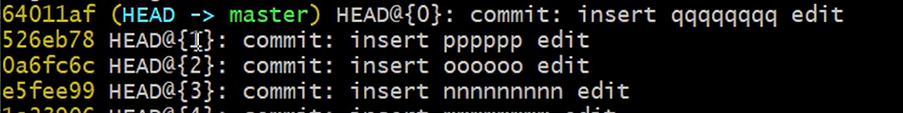
查看历史记录操作

**git log**

多屏显示控制方式

空格向下翻页，b向上翻页，q退出

git log --pretty=oneline(所有记录以一行的形式展示)

**git reflog**

**前进后退**

基于索引值操作

Git reset –hard <索引值一部分>

使用^

Git reset –hard HEAD^(向后退一步)

使用~

Git reset –hard HEAD~3(表示后退3步)

Reset的三个参数对比

Soft

仅仅在本地库移动HEAD指针

Hard

在本地库移动HEAD指针，重置暂存区，重置工作区

Mixed

在本地库移动HEAD指针，重置暂存区

永久删除文件后找回（指上传到本地库后删除了工作区的文件）

Note：不管是在工作区增删改，都要用git add来更信暂存区

回到删除那个文件之前的版本

将操作commit后，又在本地库和暂存区删除了文件，如何恢复

Git reset –hard HEAD

比较

Git diff <file>(工作区文件相对于暂存区的比较)

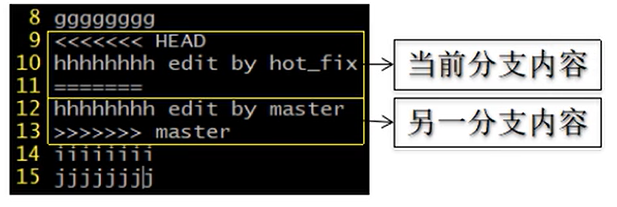
Git diff HEAD <file>(工作区相对于当前版本的本地库进行比较)

4.3分支操作

* Git branch -v(查看所有分支)
* Git branch <branch name>(创建分支)
* Git checkout <branch name>
* 合并到主分支

1. 切换到主分支
2. Git merge <另一个分支的名字>

* 解决合并冲突



在自动合并冲突后，当前分支处于正merging状态，这是可以手动合并，解决问题，也就是说将<<<HEAD,

>>>master,=====,删除；然后git add <filename>,git commit -m(没有文件名，不然会报错)，解决冲突