Uniux、Mac、Windows等几大平台上运行。 电脑上的目录 工作区 (Working Directory) 工作区中有一个隐藏的.git目录,这个不属于工作区,这个是版本库(注意:这个隐藏 的.git目录我们不要手动去改动它),版本库包含了暂存区和其他分支两部分。 git add 操作后就会把工作区的文件添加到暂存区 暂存区 (Stage) 版本库 (Reponsitory) git commit 操作后就会把暂存区的文件提交到版本库的当前分支上 git status Git会为我们自动创建一个默认主分支master,以及一个指向master分支的指针HEAD 主分支(Master Branch) 添加工作区文件的修改到暂存区 git add <文件> 其他分支。。。 git add. 一工作区 git commit -m'本次提交的描述内容' Git仓库结构 git diff <文件> stage master add 查看从最近到最远的日志信息列表 F git reset - -hard HEAD^ git reset - -hard HEAD^^ git reset --hard Git管理结构图 git reset - -hard~100 1、首先修改工作区的文件 版本库 —— HEAD git checkout -- <文件> (注意- - 之 stage master 间有空格,其中 -- 不能省略, 如果 git add 没有 -- 就变成切换分支的命令了) readme.txt LICENSE LICENSE git branch git branch <分支名字> git checkout <分支名字> git checkout -b <分支名字> 2、通过git add <文件> 把工作区文件的修改添加到暂存区 git branch -d <分支名字> Git内部操作的流程(面试重点) 一工作区 — git merge <分支名字> stage 生成ssh key git commit readme.txt window: readme.txt LICENSE 找到生成的密钥 LICENSE linux&mac 登录github账号 创建远程仓库 3、通过git commit -m '注释'就可以一次性的把暂 关联远程仓库(关联成功一次后就不 存区的文件所有修改提交到master分支上 需要在关联了) 推送到远程仓库 关联远程仓库GitHub

```
能记录每次文件的改动(包括每次改动的时间、作者、内容
什么是版本控制系统?
                 等),还可以协同其他人协作编辑的一个工具(系统)。
                 集中式版本控制系统(代表: CVS、SVN)
                                                 版本库放在中央服务器,必须联网才能工作(记录每次的改动)
                 分布式版本控制系统(代表Git)
                                        每个人的电脑就是一个完整的版本库,没有网络也可以工作(记录每次的改动)
          开发Linux系统的那个人用C语言写的Git,最开始Git只能运行在Linux
          和Unix系统上,后来移植到了Windows上,现在Git可以在Linux、
         Linux、Mac这里忽略。。日后去公司需要安装的话百度一下。
        Windows安装也比较麻烦,不过还好有一个Git面向Windows的打包好的安装程序。(Git-2.10.2-64-bit.exe)
        默认安装好后,在开始菜单中找到'Git'-》'Git Bash'即可通过Linux命令来操作电脑,从而使用Git。
                                                 git config --global user.name '你的用户名(英文)'
       配置全局的用户和邮箱(让别人知道这个是谁提交的,配置一次即可)
                                                 git config --global user.email '你的邮箱'
                创建版本库,在要使用Git管理的目录下执行这个命令,会把当前目录变为Git可以管理的仓库
                 查看Git仓库的状态(是否有文件被改动、是否有未提交等)
                              添加工作区所有文件的修改到暂存区
                                   将暂存区文件的修改提交到某个分支中(默认是master主分支)
                    查看和版本库中的不同(在未提交之前使用,方便对比自这次做了什么修改,以免提交错误)
                git log --pretty=oneline 省略日志其他信息,在一行显示
               git reflog 获取操作的版本号(一般用于回退到某个版本使用)
                      版本回退(把已经提交过的修改,回退到指定的某一个版本)
                                         回退到上个版本
                                          回退到上上个版本
                                       回退到上100个版本
                      git reset - -hard '提交ID(版本号)'
                                              回退到指定版本
                                                                  1、没有add到暂存区的时候直接使用
                                                                                            通过git checkout - - <文件>丢弃工作区的修改
                                                                                               先通过git reset HEAD <文件> 命令把暂
                                 撤销修改分两种情况(也可以通过该命令找回工作
                                                                                               存区的修改撤销掉,重新放回工作区
                                 区已删除的文件, 因为删除也是修改的操作)
                                                                  2、已经add到暂存区的时候,但还没有提交
                                                                                               在通过上一条命令丢弃工作区的修改
                                                                                               git checkout - - <文件>
                            查看分支列表以及当前所在分支
                                   创建新分支
                                    切换分支
                                      创建并且切换分支 (以上两部操作合二为一)
                                     删除分支
                                   合并某分支到当前分支
                 总结:以后开发,我们做某个功能的时候尽可能的在分支上去做,然后在分支上
                 进行版本管理,最后做完后,提交到主分支,并且推送到服务器中
                            ssh-keygen -t rsa -C 'youremail@example.com'
                                        C: Users/Auser/.ssh
                                        open ~/.ssh
                               点击头像下的Settings
                                                然后在左侧找到SSH and GPGKey,点击进入
                              新建立一个ssh key,输入title,把生成的密钥拷贝进来,保存
                            点头像左侧的小+号, New repository
                                        git remote add origin <a href="mailto:git@github.com">git@github.com</a>:yourname/gitTest.git
                             git push -u origin master
                以后每次在本地操作后,都循环去执行以上的git相关操作(add 、commit等),最后push到远程仓库
                可以通过git clone 远程仓库地址 把代码下载下来并且生成git环境(将来去公司
                后,一般来说公司已经有存在的项目仓库了,来公司后只需要git clone 下代码就可
                         git pull 把远程仓库的代码更新下来(多人合作的时候经常使用,把别人提交上去的代码,pull下来)
                更新代码
                         1、以后去公司开发,养成好习惯,每天在早上上班之后先要git
                         pull(更新服务器上最新的代码),然后在进行开发。
                         2、在开发过程中,尽量避免两个人做相同的模块,如果做相同模块的
                        话,注意代码不要乱写,每个人只写在自己的功能那里,这样以后互相
```

没有版本控制系统之前。。比如写小说、论文的痛

更新代码的时候不会造成麻烦。

苦,需要手动的去记录每次的改动。