**Git使用总结**

Git使用总结

奚振华

**摘要**

[在此处键入文档摘要。摘要通常为文档内容的简短概括。]

目录

一、git的安装 3

二、git的基本操作 3

2.1 创建版本库 3

2.2 把文件添加到仓库（暂存区） 3

2.3 把文件提交到仓库（版本库） 4

2.4 查看文件状态 $ git status 4

2.5 查看当前文件与之前文件的区别 4

2.6 显示提交日志 4

2.7 记录每一次输入的命令 4

2.8 回退版本 4

2.11 关联远程仓库 5

2.12 从远程库克隆 5

2.13 创建与合并分支 5

2.14 解决冲突 5

三、git的操作技巧 7

一、git的安装

git教程：

* http://www.liaoxuefeng.com/wiki/0013739516305929606dd18361248578c67b8067c8c017b000
  1. .1 Mac 下：

在命令行中输入git命令，会提示不存在，并会提示安装Xcode跳转到APP Store下载，安装完Xcode之后，git工具就存在了。

1.1.2 Mac 下：

通过【http://code.google.com/p/git-osx-installer/】下载，但Google地址被封，不做描述。

1.2 Window下：

….

全局变量配置

* $ git config --global user.name “Your Name”
* $ git config --global user.email “email@example.com”
* $ git config --global color.ui “always”

解决中文乱码问题

见脑图；

通过ssh链接GitHub网站

创建SSH Key：$ ssh-kengen -t rsa -C “youremail@expamle.com”

如果一切顺利的话，可以在用户主目录里找到.ssh目录，里面有id\_rsa和id\_rsa.pub两个文件，这两个就是SSH Key的秘钥对，id\_rsa是私钥，不能泄露出去，id\_rsa.pub是公钥，可以放心地告诉任何人。

二、git的基本操作

2.1 创建版本库

* $ git init

2.2 把文件添加到仓库（暂存区）

* 添加单个文件 $ git add readme.txt
* 添加多个文件 $ git add file1.txt file2.txt
* 添加同一后缀的文件 $ git add \*.txt
* 添加所有文件（包括子目录，但不包括空目录） $ git add .

2.3 把文件提交到仓库（版本库）

* 提交所有文件 $ git commit -m “wrote a readme file”
* 提交所有文件 $ git commit –m “wrote a readme file” -a (add并commit)
* 提交单个文件 $ git commit –m “wrote a readme file” readme.txt
* 增补提交（不会产生新的提交历史记录） $ git commit -C head -a --amend

2.4 查看文件状态 $ git status

2.5 查看当前文件与之前文件的区别

* $ git diff readme.txt

备注：+是当前文件新增的，-是之前文件有，当前文件不存在的

2.6 显示提交信息/日志

2.6.1 $ git show 查看上次提交的文件信息

2.6.2 查看文件提交日志

2.6.2.1 $ git log 详细

2.6.2.2 $ git log --pretty=oneline（简单，一行显示一条日志记录）

2.6.3 $ git reflog （查看reset之前的文件提交日志，可以用来回退未来版本）

2.7 暂时空

2.8 回退版本

* $ git reset --hard 3628164 把本地仓库回滚到指定HEAD
* $ git reset --hard origin/[分支名] 把本地仓库回滚到指定远程分支

备注： commit id 不需要写全，git会自动搜索

2.9 将工作区的修改全部撤销

$ git checkout -- readme.txt

将文件回到最近一次git commit 或 git add 时的状态

备注：在暂存区的修改撤销，使用 $ git reset命令

2.10 版本库中删除文件

$ git rm test.txt ，然后再提交 $git commit -m “remove test.txt”

2.11 关联远程仓库

2.11.1 查看本地分支与远程分支对应关系

$ git remote show origin

2.11.2 要关联一个远程库，使用命令：

$ git remote add origin git@server-name:path/repo-name.git

2.11.3 关联后，使用命令 $git push –u origin master 第一次推送master分支的所有内容

2.11.4 此后，每次本地提交后，只要有必要，就可以使用命令：

$ git push origin master 推送最新修改

2.11.5 同步指定本地分支到远程指定分支

$ git push origin [本地分支]:[远程分支]

2.12 从远程库克隆

2.12.1 登陆GitHub，创建一个新的库

2.12.2 使用git clone克隆一个本地库：

$ git clone git@github.com:xizhenhua/clone.git

2.13 创建与合并分支

2.13.1 切换到本地制定分支：$ git checkout [分支名]

2.13.2 创建分支：$ git checkout -b dev （把当前本地分支克隆为本地新分支）

等同于：$ git branch dev （创建dev分支）

$ git checkout dev （切换到dev分支）

2.13.3 创建新分支，把指定远程分支克隆成本地新分支

$ git checkout -b [分支名] origin/[远程分支名]

2.13.4 查看当前分支：$ git branch

会出现：\* dev

master

前面有\*号代表，目前切换到得当前分支

查询所有分支信息（包括远程） $ git branch -a

2.13.5 合并分支：$ git merge dev

2.13.6 通过--no-ff参数，表示禁用Fast forward

$ git merge --no-ff -m “merge with no-ff” dev

因为合并要创建一个新的commit，所以加上-m参数，把commit描述写进去。

这样就能看到合并合并分支的历史，否则通过默认Fast forward模式合并是看不到合并分支的历史。

2.13.7 删除分支：$ git branch -d dev

2.14 解决冲突

2.14.1 提示自动合并失败 $ git merge feature1

*Auto-merging readme.txt*

*CONFLICT (content): Merge conflict in readme.txt*

*Automatic merge failed; fix conflicts and then commit the result.*

2.14.2 查看合并失败原因 $ git status

*# On branch master*

*# Your branch is ahead of 'origin/master' by 2 commits.*

*#*

*# Unmerged paths:*

*# (use "git add/rm <file>..." as appropriate to mark resolution)*

*#*

*# both modified: readme.txt*

*#*

*no changes added to commit (use "git add" and/or "git commit -a")*

2.14.3 查看冲突文件

*Git is a distributed version control system.*

*Git is free software distributed under the GPL.*

*Git has a mutable index called stage.*

*Git tracks changes of files.*

*<<<<<<< HEAD*

*Creating a new branch is quick & simple.*

*=======*

*Creating a new branch is quick AND simple.*

*>>>>>>> feature1*

Git用<<<<<<<，=======，>>>>>>>标记出不同分支的内容，我们修改如下后保存。

2.14.4 找到冲突修改完之后，进行$ git add 和 $ git commit命令进行提交。

2.14.5 查看分支合并的情况

$ git log --graph --pretty=oneline --abbrey-commit

*$ git log --graph --pretty=oneline --abbrev-commit*

*\* 59bc1cb conflict fixed*

*|\*

*| \* 75a857c AND simple*

*\* | 400b400 & simple*

*|/*

*\* fec145a branch test*

*...*

2.14.6 最后删除分支

2.15 分支管理策略

2.15.1 通过--no-ff参数，表示禁用Fast forward

$ git merge --no-ff -m “merge with no-ff” dev

因为合并要创建一个新的commit，所以加上-m参数，把commit描述写进去。

这样就能看到合并合并分支的历史，否则通过默认Fast forward模式合并是看不到合并分支的历史。

三、Git的操作技巧

3.1 快速拉取所有分支

3.1.1 $ git clone git\_url创建分支(克隆远程仓库，默认是拉去master分支)

3.1.2 $ git fetch --all (抓取远程服务器所有分支)

3.1.3 $ git reset --hard origin/[分支名] 把本地仓库回滚到指定远程分支

Github网站

Fork功能：意思是将别人的版本库克隆到你的账户下，从而能就行随意修改

推送pull request功能：就是将克隆了别人项目的代码就行修改之后，再推送给对方，至于对方接不接受是另外一回事了。

repository 版本库又名仓库

工作区（Working Directory）

tracked 跟踪

暂存区（Stage）

冲突 conflict

Changes not staged for commit:

提交未被暂存的变化内容。

(use "git add <file>..." to update what will be committed)

使用“git add <file>...”这个命令去更新所要提交的内容

(use "git checkout -- <file>..." to discard changes in working directory)

在工作目录使用"git checkout -- <file>..."这个命令，丢弃工作区所做的修改

nothing to commit, working directory clean

没有需要提交的修改，而且，工作目录是干净的。

GitHub账户

用户名：xizhenhua

邮箱：61654367@qq.com

密码：123qazxdr

ssh密码：123qazxdr

github.com

HTTPS：https://github.com/xizhenhua/www.git

SSH：git@github.com:xizhenhua/www.git