学习笔记

# Git和svn对比

1. 速度

在很多情况下，git的速度远远比svn快(可以本地提交，不用联网)

1. 结构

Svn是集中式管理（所有代码的版本控制都在服务器），git是分布式管理（服务器和客户端都存在代码的版本）

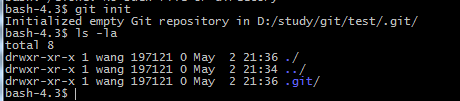
1. 其他

Svn使用分支比较笨拙（分支和tag是拷贝的形式，浪费空间和时间），git可以轻松拥有无限个分支（分支和tag,只是打个标记，不用拷贝代码）

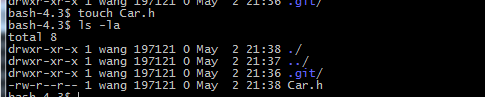
Svn必须联网才能正常工作（企业中离了公司无法进行svn操作），git支持本地版本控制（离开了公司依然可以开发）

# 常用命令

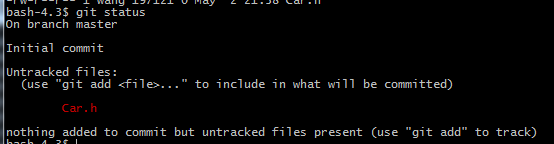
1. 查看帮助：git help
2. 查看指定命令：git help clone
3. 搜索关键字：/clone（翻页：F下一页 B上一页）
4. 退出帮助文档：输入Q
5. Svn checkout -> git clone检出
6. Svn update -> git pull 更新
7. Svn commit -> git push 提交
8. Git commit 提交(不同于git push，git commit提交到本地没有提交到服务器)
9. 初始化git本地仓库：git init



1. 创建文件Car.h：touch Car.h

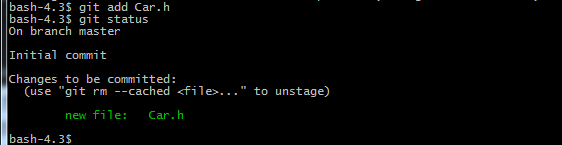


1. 查看文件状态：git status



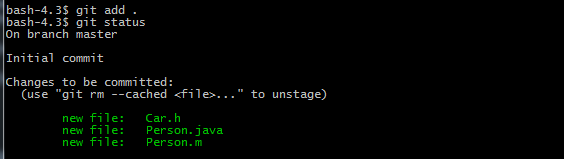
红色文件：新增文件没有被纳入git管理中，或者文件被修改了没有提交

1. 添加文件到git暂存区：git add Car.h



绿色文件：文件加入到了“暂存区” （修改完文件之后也需要先执行add操作）

1. 添加多个文件到git暂存区：git add file1 file2 … filen
2. 添加多个文件到git暂存区：git add . (“.”表示当前目录下的所有新创建的文件)



1. 提交：git commit –m “提示信息”
2. 查看日志：git log
3. 打开文件：open file
4. 查看配置文件：git config -l
5. 编辑配置文件：git config -e

# 配置账号信息

配置局部信息：保存在.git下的config中

1. 配置用户名：git config user.name “wangpiece” （区分谁开发的）
2. 配置邮箱信息：git config user.email “wangpiece.com” （用于联系开发人）

配置全局信息：保存在C:\Users\wang\.gitconfig中

1. 配置用户名：git config --global user.name “wangpiece” （区分谁开发的）
2. 配置邮箱信息：git config --global user.email “wangpiece.com” （用于联系开发人）

使用优先级：局部->全局

# 查看日志

1. 查看日志：git log
2. 版本号：40位，哈希值（唯一）

# 起别名

1. 配置局部命令别名：git config alias.st “status” 🡪点后面的st表示命令的简写，双引号里面的为原指令的全称
2. 配置局部命令别名(带参数)：git config alias.ci “commit -m”
3. 全局配置在config后面加--global即可

# 版本回退

文件已经被commit:

1. 版本回退：git reset --hard HEAD^(回退到上一个版本，一个^表示一个版本)
2. 回退到指定：git reset --hard 版本号的前7位

文件没有被commit:

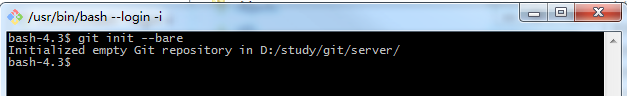
1. 回退到当前最高版本：git reset --hard HEAD
2. 检出当前分支的最高版本：git checkout Car.h

# 查看修改参考日志

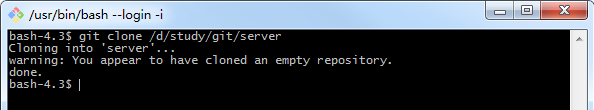
1. 如果git回退到早期版本，那么后面的那些版本健不存在，如果此时想要回退到之前的时候，那么可以通过git reflog来查看

# 多人开发练习

1. 初始化服务：git init –bare



1. 克隆代码：git clone 路径



1. 设置忽略文件

