## 燕十八git教程

安装 : command prompt 命令提示符

terminal emulator 终端仿真程序

console window 控制台窗口

custom setup 自定义安装

黑马git教程笔记

git是一个分布式版本控制工具，他与svn不同的是，git有一个本地服务器，有本地仓库，同时可以连接到远程服务器，比如github

在git中，每个用户的电脑本地都有这个项目的完整信息，可以完全脱离网络服务器来独立使用，网络服务器只是本地的一个副本

也就是说git相当于有两个服务器，一个是每个开发者独有的本地服务器库，另一个是在网络上的服务器库（一般用于共享代码），本地用户完成了一个项目，commit到本地服务器，然后在把本地服务器上的所有代码push到远程的服务器上

git的特点：

分布式，强调个体

公共服务器压力和数据量都不会太大 如果是svn 所有开发者都要连接到一个svn服务器

速度快、灵活 本地仓库读取，自然很快

任意两个开发者之间可以很容易的解决冲突

离线工作

每日工作备份

可以吃后悔药

开源项目的工作流程：人物：主开发者 普通开发者 公共服务器 每个开发者本地服务器

权限：主开发者可以往公共服务器push权威代码

普通开发者只能通过pull拉取公共服务器上由主开发者提供的代码，不能直接往公共服务器提交代码，如果确实要，只能把补丁发给主开发者，审核，在决定是否push到公共服务器

普通开发者之间也是通过发补丁的方法解决冲突

git的常用命令：

1. 创建git资源库

git init --bare 库名称

2. 在用户文件夹下把资源clone下来

git clone <仓库目录> /g/software/repository/git/itheima <用户目录> .

注意：clone时候用户目录必须为空

3. 创建一个文件，纳入到版本控制中。

git add <文件名>

第一次执行会出警告：warning: LF will be replaced by CRLF in readme.txt.

The file will have its original line endings in your working directory.

警告处理：需要配置用户信息

git config user.name = "zhangsan"

git config user.email = "itheima@gmail.com"

4. 提交到本地版本库里。

git commit <文件名>

5. 推送到远程共享版本库中

git push origin master origin为远程库URL地址，master为本地主分支

6. 切换用户，拉取最新的文件，也就是拉去别人提交到服务器上的最新文件

git pull

git与svn的命令对比：

git命令 svn命令

clone checkout

pull update

push commit

commit

git pull 之后有冲突:

$ git pull

remote: Counting objects: 5, done.

remote: Compressing objects: 100% (2/2), done.

remote: Total 3 (delta 0), reused 0 (delta 0)

Unpacking objects: 100% (3/3), done.

From g:/software/repository/git/itheima26

50db030..a04027a master -> origin/master

Auto-merging readme.txt

CONFLICT (content): Merge conflict in readme.txt

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

解决方法:

1.git mergetool

调用上面命令之后会提示你输入编辑方法,输入:beyond compare 进入编辑器.解决冲突时只要编辑红色部分，确定要保存那部分内容

2.编辑完之后.调用git commit -a 把当前目录中所有都提交到本地库中.

3.git push origin master 提交到远程库中.

由于tortoisegit和tortoisesvn是同一个公司出的，所以，操作雷同

只是该类对应的命令 clone（首次使用git时，签出服务器的所有代码） pull（相当于update本地数据为服务器最新版本） push（推送到服务器） commit（提交到本地）

解决冲突的方法也是一样的，

当push到服务器提示冲突时，使用pull拉取服务器最新版本回来，此时又提示冲突，然后编辑冲突edit conflict 保存 标记为resolve 然后提交 提交完后再push

git很只能，以上操作都是一环扣一环的！！！

如何打包补丁，合并补丁呢？

在目录下点击 create patch serial

普通开发者之间可以通过邮件发送补丁，当收到补丁后，放到指定目录，apply patch应用补丁，此时又生成新的合并过后的补丁，然后再通过邮件发送给主开发者，让他审核，判断是否推送到公共服务器上

注意：如果有多个补丁，在打补丁的时候一定要按照先后顺序，因为有些补丁是建立在其他补丁的基础之上的

如果本地文件和补丁文件有冲突，是不能合并补丁的，此时又回到编辑冲突了（注意先不关闭应用补丁窗口，解决冲突后再关），解决完冲突后，再次应用补丁即可

*我的开源中国上的账号 1780858508@qq.com pwd:zmh123*

*我的github账号：username:zmh19941223 pwd:zhuminghao123 也可以用邮箱登录，我的qq邮箱*

*github账号2：username:zmhtest 15218401303@163.com pwd:zhuminghao123*

*官网：*[*https://github.com/*](https://github.com/)

**传智播客github详细教程笔记**

在gitbash中利用命令行的方式创建一个版本库，克隆远程github上的

echo "# yii2" >> README.md

//往个readme文件写入内容，如果不存在文件则创建，在写入，这是markdown的写法

git init

//初始化本地库，产生一个.git隐藏文件夹，这是本地master分支

git add README.md

//添加到暂存区

git commit -m "first commit"

//提交到本地版本库里

git remote add origin https://github.com/zmh19941223/yii2.git

在本地库的 config文件中加了：

[remote "origin"]

url = https://github.com/zmh19941223/yii2.git

fetch = +refs/heads/\*:refs/remotes/origin/\*

这样，origin就代表URL地址下的远程库

git push -u origin master

把本地的分支master推送到远程仓库

此时，会弹出一个窗口，要求输入github的用户名和密码

如果当前分支与多个主机存在追踪关系，则可以使用-u选项指定一个默认主机，这样后面就可以不加任何参数使用Git push。

注意：只要本地目录用 git init生成了.git文件夹，无论是用gitbash 或者用 小乌龟 都是可以识别的，也就是都可以修改，读取.git 下的配置文件...所以可以正常提交commit，push，推送到远程！！！

什么是pull request呢？也就是两个开发者 交流，合并分支

有一个仓库，叫Repo A。你如果要往里贡献代码，首先要Fork这个Repo，于是在你的Github账号下有了一个Repo A2,。然后你在这个A2下工作，Commit，push等。然后你希望原始仓库Repo A合并你的工作，你可以在Github上发起一个Pull Request，意思是请求Repo A的所有者从你的A2合并分支。如果被审核通过并正式合并，这样你就为项目A做贡献了

注意：使用哪个github账号登录，就代表你有权限修改哪个用户的信息，你可以通过你的账号访问任何人的github仓库，只要你有他们仓库的地址即可

fork操作：把别人的整个项目全部拷贝到你的一个新的仓库中，一个项目对应一个仓库

fork后仓库是独立存在的

watch：意思即是查看 关注 订阅，当你关注的项目有了更改，就会显示在你的github首页，也就是动态

starred：收藏你喜欢的项目

### svn版本控制

首先下载subversion 服务器端

安装好后他会在环境变量中自动添加路径，所以直接在命令行就可使用svn svnadmin svnserve 等命令

接下来就是创建资源库repository，新建一个文件夹，下面有若干文件夹（用来放多个资源库），用svnadmin create 目录名 //这样会生成一堆文件夹，开启svn服务

1、运行命令行，输入svnserve -d -r 资源库目录名

2、将svnserve注册为系统服务，并设置自动启动

方法为：管理员身份运行命令行 输入 ：sc create svn binpath= "\"C:\Program Files（86）\Subversion\bin\svnserve.exe\" --service -r G:\svnhome" displayname= "Subversion Server" depend= Tcpip start= auto

出现 CreateService 成功 即可！

讲解:sc(注册服务)

create svn（服务的名称）

binpath= （这里特别注意=前面不能有空格，后面要有一个空格）（服务二进制文件的路径）"\"D:\Program Files\Subversion\bin\svnserve.exe\" --service -r G:\svnhome"（设定资源库的根目录，方便访问其他资源库） displayname= "Subversion Server"（这是在服务列表中显示的名称）

depend= Tcpip start= auto（基于Tcpip 自动启动）

然后执行以下配置：

在配置eclipse插件svn的时候，需要配置一下内容才能成功连接svnserve.conf:

[general]

anon-access = read // 这里是设置匿名访问如果提示条目不可读，只需将这里设置成none即可

auth-access = write //授权访问 给特定的用户指定可以访问的资源

password-db = passwd //表示采用本目录下的哪个文件来保存密码

authz-db = authz //表示用当前目录下的authz文件来保存授权信息

passwd: //可以在这里创建多个用户和密码

[users]

### 这里是设置用户的密码，用于以后登录，可以配置多个组员的

harry = harryssecret

admin = 123

ceshi = 123

authz:

[groups] //这里用来创建组

group1 = admin,harry

### [/]表示根目录下的所有资源

[/]

### 给harry用户读和写的权限

harry = rw

ceshi = r //只有读权限

@group1 = rw //表示这个组具有读写权限

\* = //起到屏蔽作用，表示出了以上用户，其他人什么权限都没有

特别注意，前面是不能有任何空格的！###表示注释！

详细教程：http://jingyan.baidu.com/article/48206aeada1741216bd6b34e.html

过时和冲突：也就是多个组员修改了同一个地方，其中一个组员提交了，后来另外一个组员在没有更新时就提交，就会出现冲突，提示后一个组员的已经过时了。解决办法：编辑冲突，然后点击标记为解决 即可！

\* 号表示有比服务器更新的修改，这是重新提交即可，然后在其他组员上检出。

解决冲突的最好方法是：先更新一下，取得服务器上最新版本，然后在此基础上做修改。

读：表示只能从服务器检出数据；

写：表示把数据提交到服务器。一般给测试人员提供读的权限，给开发者提供读和写权限

时光机（代码还原）：

**svn燕十八教程笔记**

首先安装svn的客户端tortoisesvn

在客户端即可完成，建仓库，把内容提交到仓库

他是一个集中式管理代码的软件

使用svn的好处：

1.方便团队协同开发

2.版本控制，当程序出现问题是可以轻松回退到之前的版本

3.可以清楚记录是哪个开发者修改了哪个部分，分清责任

开发不同的文件，如何把文件整合到一块，开发相同文件，如何解决冲突，如何回退？

svn工作原理：

有一台svn服务器，里面有一个仓库

团队开发人员开发不同的模块，提交到仓库中，这个仓库会记录各个版本的信息

团队来了新成员：首先要 checkout 签出仓库中的代码

先更新 再提交 这样是为了防止冲突 如果有冲突是不能正常提交的

在开始书写代码之前，必须要先更新，更新的作用是把服务器上最新的代码拿过来，在最新代码的基础上进行开发

更新 干活 更新 提交

注意：更新是指把服务器上的最新代码更新到本地，然后再合并本地你写好的代码，提交

本地仓库练习

步骤；

1、建一个新的目录

2、右键算则 create repository here

3、建完就走 记住访问路径 比如 file:///D:\www\svn

解决文件图标不显示的方法：

1在程序的文件夹下，右键--TortoiseSVN--Settings(设置)得到图1，然后弹出如图2所示对话框，点击Icon Overlays--Status cache,选择Shell--确定。

2找到SVN安装包，双击打开SVN安装包，点击Next--选择Repair（修复）--Repair,修复完成后，重新启动系统。

删除一个文件的步骤：不能简单通过普通删除 要使用tortoise右键

delete来删，删完后，更新，提交

对于一个全新的文件首先要把他 add 到版本控制，此时图标由 ？-> + 然后更新、提交

可以通过右键 选择 repo browser 查看各个版本的文件内容，然后再通过update to reversion命令更新到指定的之前版本

在提交界面一定要写 备注 说明更新或修改了哪一部分，这些备注可以通过 show log来查看

签出界面选项详解：

checkout depth：签出的程度

full recursive ：全部递归签出，也就是签出所有

immediate children：直接的子元素

还可以选择choose item 选择签出哪个文件

在show log中可以看到整个项目的所有版本信息，看完再决定签出那个版本

不同的开发者修改了文件的同一个地方，于是出现冲突，提交不了

冲突提示：

Modified: E:\svnTest2\demo.txt

Error: Commit failed (details follow):

Error: File 'E:\svnTest2\demo.txt' is out of date

Error: File '/demo.txt' is out of date

Error: You have to update your working copy first.

Completed!:

有冲突的更新是这样的：

如果是更新呢？

Command: Update

Updating: E:\svnTest2

Conflicted: E:\svnTest2\demo.txt

Completed: At revision: 5

Warning!: One or more files are in a conflicted state. 冲突状态

此时，生成一堆？？？文件，需要你手动解决冲突

<<<<<<< .mine

hello

world||||||| .r4

hello======= 版本分割线

hello

世界>>>>>>> .r5

手动解决

hello

world

右键选择 resolved 确定，自动删除带？文件，然后提交

在window下搭建svn服务器 subversion svn

首先安装subversion svn

建立目录 比如 svnrep/apple orange

在目录中建仓库 svnadmin create svnrep/apple

此时，仓库生成很多文件，有 conf data

配置好密码数据库，目录读取权限

开启服务器 svnserve -d -r d:/svnrep 配到库的上一级即可

通过svn客户端访问：svn://192.168.1.100/apple 签出

出现认证失败是因为没有配置授权文件和密码数据文件

当已经配置了还是出现认证失败，很可能是配置的格式错了

所有的配置文件都是不允许行的开头有空格的！！！

你可以通过svnadmin create xxx 在一个文件夹下创建多个库，用于存放不同的项目

在签出时，只需指定地址即可 svn://ip地址/目录名

注意这个ip地址对应的就是各个库的上一级目录。在用svnserve -d -r xxx 开启服务器时要特别注意

在window下搭建svn服务器 visualsvn

可以选择添加svn命令行工具到系统的环境变量path

初始化服务器配置

可以设置库路径

可以自定义服务器端口 8843

使用安全连接 https://

https://zmh:8443/svn/apple

在一个目录中利用 tortoise 的checkout命令，签出，提示要用户名和密码

所以我们要在服务器中配置user：create user

然后用这个用户名和密码进行签出 作为版本一

visualsvn使用技巧：

可以在properties中设置 库根目录的位置

和通过svn客户端访问的URL地址，这些都是可以通过设置server name 和端口来重新定义的

同时还可以通过在浏览器中输入svn地址，查看库里面的内容

https://zmh:8443/svn/apple

但是要输入用户名和密码

设置了只读权限的用户只能 update 不能 Commit

提交时会报错

Command: Commit

Error: Commit failed (details follow):

Error: Access to '/svn/apple/!svn/me' forbidden

Completed!:

问题是怎样知道是哪个用户呢？

是在第一次签出时，输入的用户就是这个目录的用户

在浏览器访问时，每次都会提示输入用户密码的

但是在签出时，怎样设置输入用户呢？

原来是tortoise记住了密码，先清空 save data 中的授权信息authentication data

此时需要密码签出，也可勾选保存密码

这个是自定义的ip地址，也可以使用https://zmh:8843/svn.apple来访问

https://192.168.1.100:8443/svn/apple

如果改了文件名，svn是无法检测出来的，他会当做是一个新的文件，先要 add 在更新 提交

所以删除和重命名都只能通过tortoisesvn这个菜单中的delete rename选项，然后更新、提交 才是完成最终修改

在window中服务管理进程，进程管理线程

也就是说一个服务可以有多个进程，一个进程可以有多个线程

如果选中某个文件更新，则更新的是这个文件，如果在目录更新，那么更新的是这个目录的所有文件

**组的概念**

group 一个用户可以属于多个组，比如同时属于PHP Android组

就要在这两个组中都添加这个用户

如果有两个用户修改了同一个文件的不同行，由于提交有先后顺序，

后提交的用户直接提交会报 文件过时错误，提示更新

在更新时，可以看到，他把两个用户所修改的文件进行了合并 merge

然后可以正常commit了

Error: Commit failed (details follow):

Error: File 'D:\visualsvn\cbh\测试文件最新版.txt' is out of date

Error: File '/测试文件最新版.txt' is out of date

Error: You have to update your working copy first.

Completed!:

Merged: D:\visualsvn\cbh\测试文件最新版.txt

Completed: At revision: 15

技巧：如果将tortoisesvn中的advanced 中的allowauthSave设为false，则 每次checkout update都会要求输入用户名，密码，这样才更符合实际开发场景

冲突往往是由于没有更新就开始编辑造成的，所以要养成良好习惯，上来工作就更新，完成一个功能就更新提交

出现冲突后可以使用edit conflict方便编辑冲突

点击有冲突的行，红色部分，右键有如下选型

use this text block

use ...before theirs 选择使用谁的代码在前面

use ....before mine

修改完成后 更新 提交

update to version 2

只是从服务器的版本库中取回之前的某个版本，并不会删除服务器的最新版本

更新的到的永远是服务器的最新版本！！！

问题场景：如何让服务器最新版本会退到较早之前的版本？

在 show log 中通过 revert to this version 即可将该版本恢复到本地工作目录，提交即可覆盖原来的最新版本 他是先把本地目录清空，再从服务器版本库取回来指定版本，取回单文件带 + 号 只需再次在目录下提交即可覆盖服务器上的所有文件

update就是将服务器上的文件复制一份到本地，在哪里update 就update谁

注意：解决了冲突要马上提交到服务器，而不是先更新再提交，否则又有冲突，因为服务器的最新文件并没有改变

注意:版本库主要是用来存放不同版本变动信息，出现问题我们只需把原来的版本提取出来，放到运行环境上即可 也可以把恢复到本地的版本重新提交，此时，版本库又多了一个版本，作为最新版本

反正 只要是有修改或者添加文件的提交都会创建一个新的版本

文件和目录分别有各自的日志记录

练习：尝试用svn来管理一个java项目

情景：有一个版本一，发布后，继续开发版本二，用户反馈版本一有bug，调用部分人员去修复版本一的bug，修复后，发布，并把修复后的代码整合到版本二

步骤：1.首先创建一个版本库 这个版本库有三个目录 trunk branch tags 分别用于存放 主线文件，主要用于开发各个版本 、分支版本（用于存放tags中有问题的版本，并在此基础上进行修复）、tags用于存放最终要发布的版本，只是用来存放，一般不做修改

2.连接版本库，直接检出trunk到一个空项目，在这个空项目中开发版本一，新来开发者，只需从版本库中检出，即可开发

3.开发完版本一后，提交到tags目录，点击 “分支与标记” 选择tags文件夹，并在URL中起一个名字标记版本，比如：v1.0

4.继续在版本一基础上开发版本二，此时，版本一出现了bug，

5.在项目上点击右键，切换到tags下对应的版本一，把tags中的版本一检出到项目

6.由于tags中的文件是不能修改的，所以我们需要创建v1的一个分支

右键选择 分支与标记 选择 branch 文件夹 命名URL 确定即可

7.再次将一个项目 切换 到branch下的相应版本，在此基础上进行修改，改好后 点击 分支与标记 将修复的文件发布到tags文件夹下，命名URl

6.将修复的文件整合到版本二 回到主线trunk项目，选择文件右键点击合并，选择要合并的文件 全部选择默认选项即可 合并后，迅速提交到服务器让其他开发者能看到

**开发一个全新项目的基本流程**

创建一个项目->创建一个版本库->连接到这个版本库->把项目提交到版本库中的trunk文件夹(右击项目选择share project->svn->选择trunk文件夹，此时就会在trunk文件夹下创建以项目名命名的文件夹->finish)->在弹出的同步视图中右击项目，选择提交，选择要提交的文件

提示：在svnserver中，选择某个库，右键选择 copy URL to ...将这个库的地址复制到剪贴板

问题：为什么使用eclipsesvn插件连不上svn服务器？

软件不兼容问题，下载新软件进行安装

经过测试发现是svn 接口 connector 出错，不兼容，应该为 javahl xxx

如果没有，则安装此插件 http://subclipse.tigris.org/update\_1.8.x 除了 mylyn