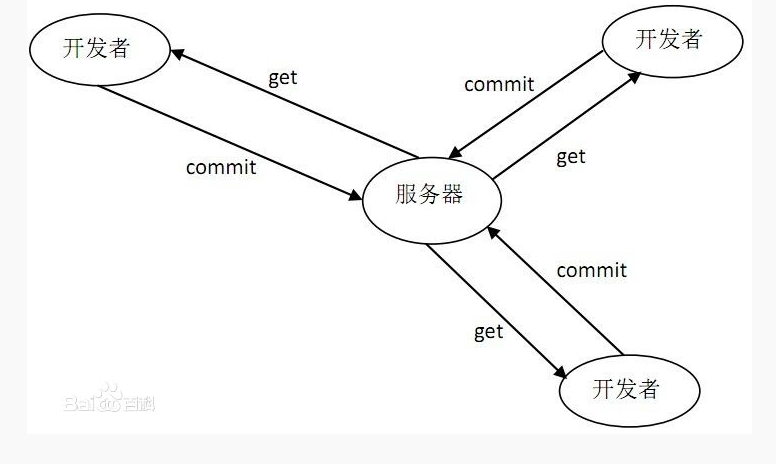
# SVN版本控制

## 1:SVN简介

svn([subversion](http://baike.baidu.com/view/429581.htm" \t "_blank))是近年来崛起的[版本管理](http://baike.baidu.com/view/3658694.htm" \t "_blank)工具，是[cvs](http://baike.baidu.com/view/20982.htm" \t "_blank)的接班人。目前，绝大多数[开源软件](http://baike.baidu.com/view/444964.htm" \t "_blank)都使用svn作为代码[版本管理](http://baike.baidu.com/view/3658694.htm" \t "_blank)软件。

svn的工作模式如下图所示



SVN采用了集中式代码管理,集中式代码管理的核心是服务器，所有开发者在开始新一天的工作之前必须从服务器获取代码，然后开发，最后解决冲突，提交。所有的版本信息都放在服务器上。如果脱离了服务器，开发者基本上可以说是无法工作的。下面举例说明：

开始新一天的工作：

1、从服务器下载项目组最新代码。

2、进入自己的分支，进行工作，每隔一个小时向服务器自己的分支提交一次代码（很多人都有这个习惯。因为有时候自己对代码改来改去，最后又想还原到前一个小时的版本，或者看看前一个小时自己修改了哪些代码，就需要这样做了）。

3、下班时间快到了，把自己的分支合并到服务器主分支上，一天的工作完成，并反映给服务器。

这就是经典的svn工作流程，从流程上看，有不少缺点，但也有优点。

**缺点**

1、服务器压力太大，数据库容量暴增。

2、如果不能连接到服务器上，基本上不可以工作，看上面第二步，如果服务器不能连接上，就不能提交，还原，对比等等。

3、不适合开源开发（开发人数非常非常多，但是Google app engine就是用svn的）。但是一般[集中式管理](http://baike.baidu.com/view/3794518.htm" \t "_blank)的有非常明确的权限管理机制（例如分支访问限制），可以实现分层管理，从而很好的解决开发人数众多的问题。

**优点**

1、管理方便，逻辑明确，符合一般人思维习惯。

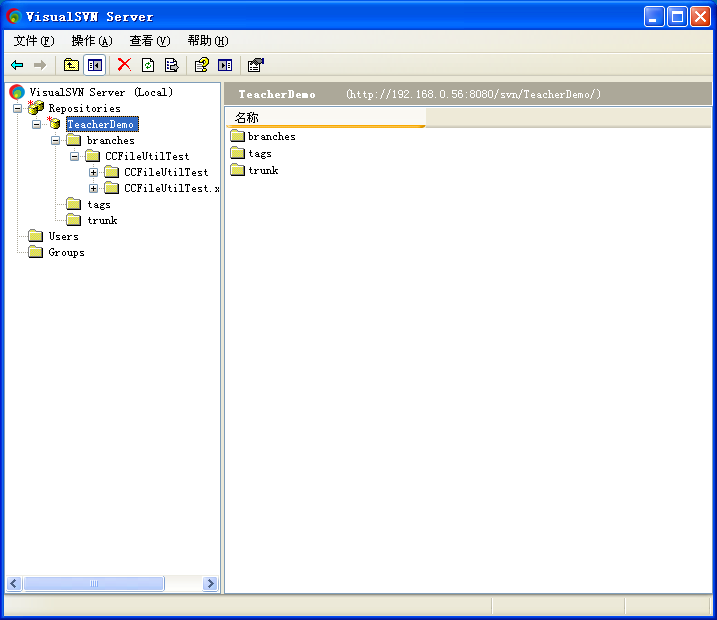
2、易于管理，集中式服务器更能保证安全性。

3、代码一致性非常高。

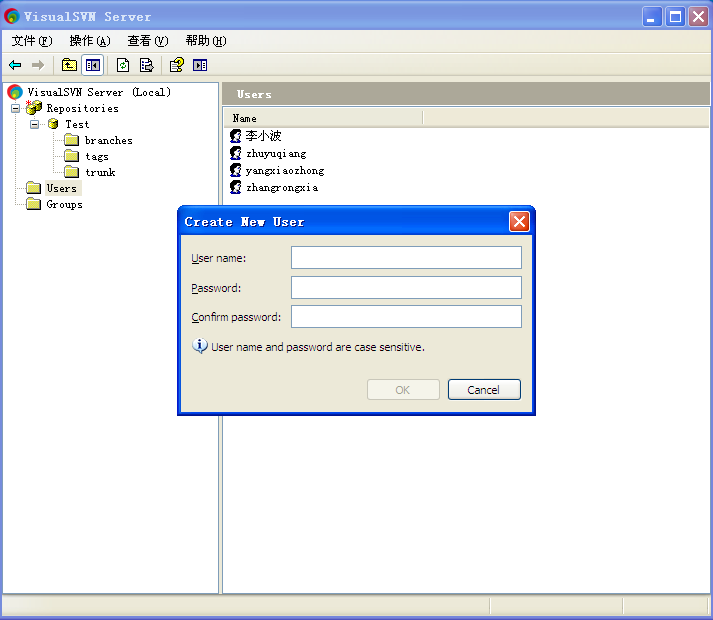
4、适合开发人数不多的项目开发。

## 2 :SVN服务器的配置

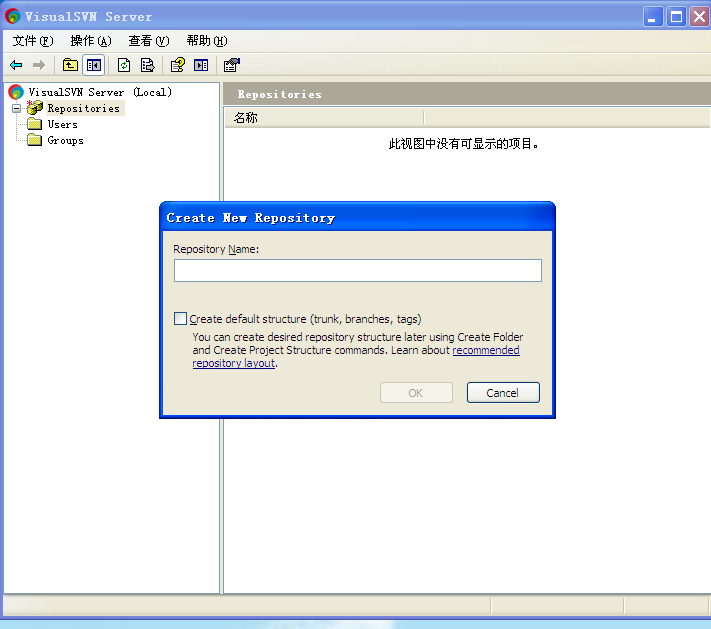
(1)首先下载VisualSVN Server,我们这里是在xp环境下进行配置,下载完成之后进行安装,如下图所示。可以看到VisualSVN Server下有三个文件夹,第一个Repositories是代码数据仓库，第二个Users是用户，第三个Groups是组。



(2)添加用户,右击User,创建用户,看到如下界面,我们指定用户名和密码,就成功创建了用户

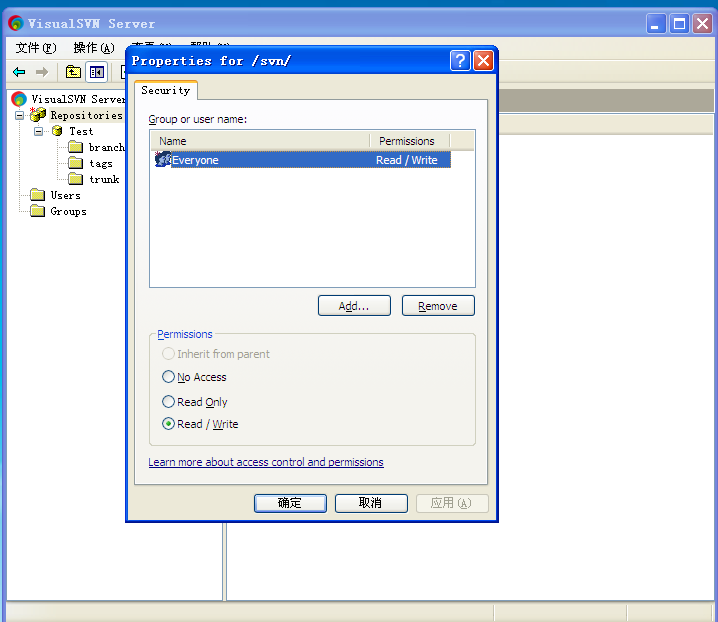


(3)创建新的代码仓库,右击Repositories,Create New Repositories,出现如下所示界面

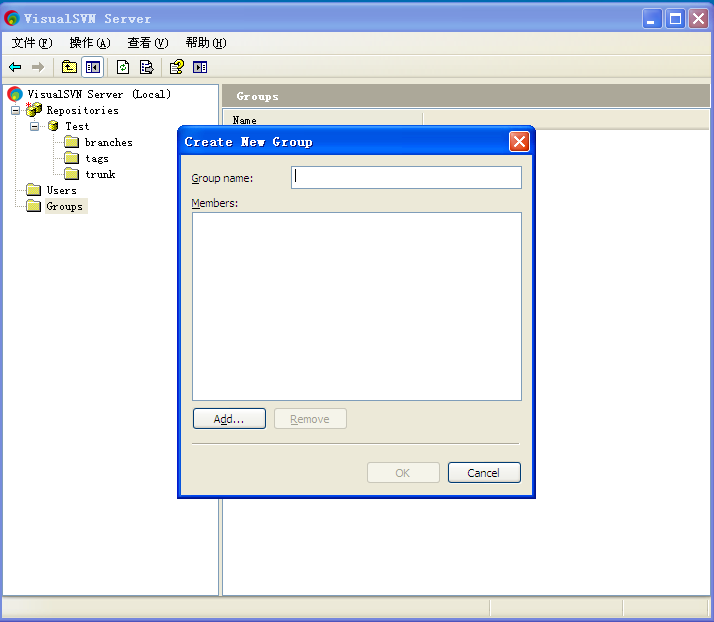


然后我们输入代码仓库的名字，这样一个新的代码仓库就创建成功，下面我们给这个代码仓库分配权限

（4）权限分配

右击Repositories,选择Properties,出现如下所示界面,我们就在这个界面中对我们的用户或者组员来分配权限,权限分为三种:无权限,只读,和读写,我们在这里选择第三种.,如果其他用户也需要添加权限,我们选择Add,然后给对应的人分配权限即可.

(5)创建小组

右击Groups,创建组,如下图所示,我们可以直接在组里面添加用户,并且可以通过小组来分配权限,节省时间.

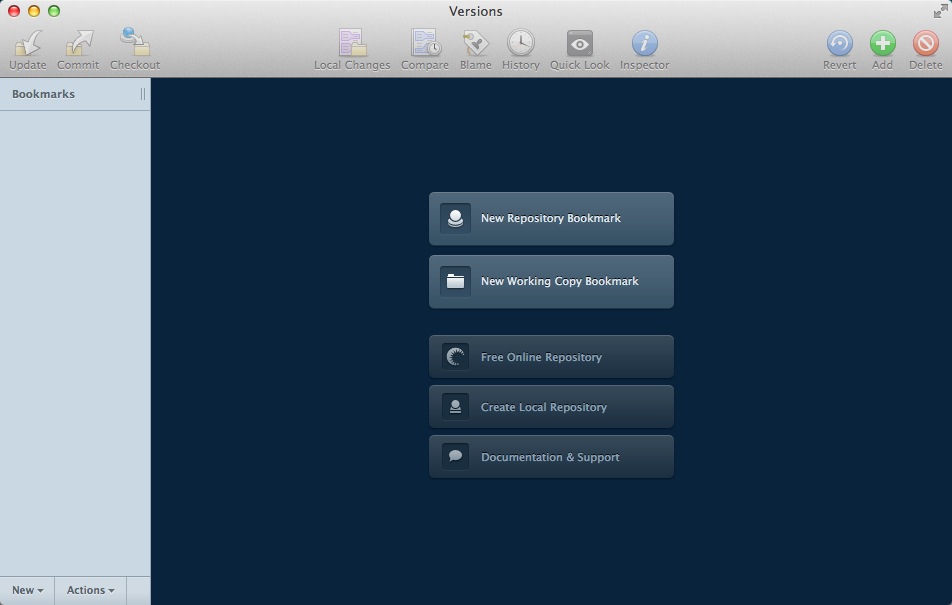
这样服务器端最基本的配置就完成了,我们只需要在客户端根据我们svn服务器的网址来访问即可,在这里我们的服务器是

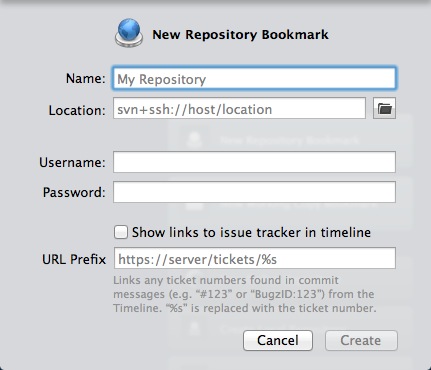
<http://192.168.0.56:8080/svn/Test>，其中Test就是我们新建的代码仓库的名字。

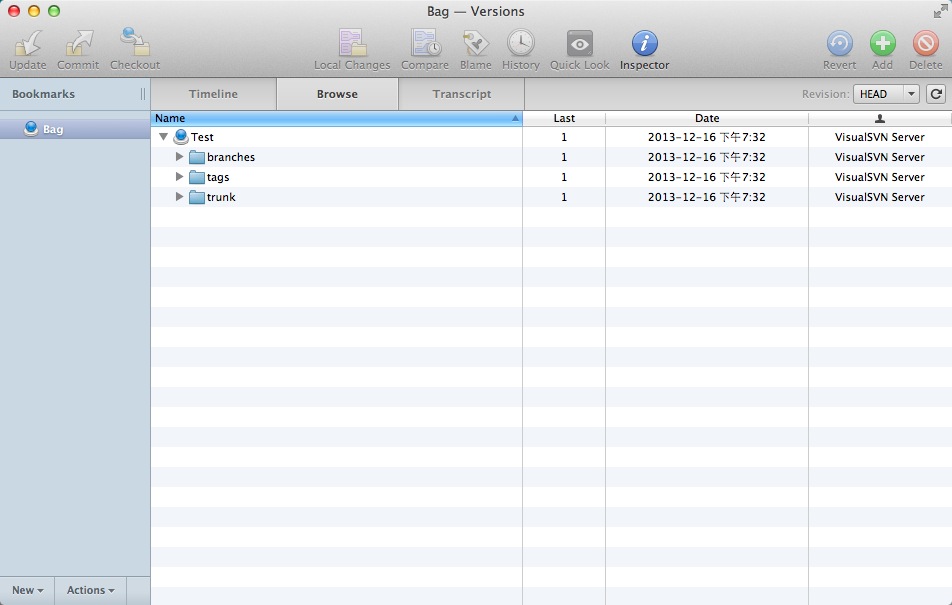
## 3:Mac下配置SVN客户端

上一节我们讲了服务器端的配置，在这一小节我们讲解一下客户端的配置（我们在这里用的是MAC机器，所以我们就以MAC客户端为例子讲解一下）

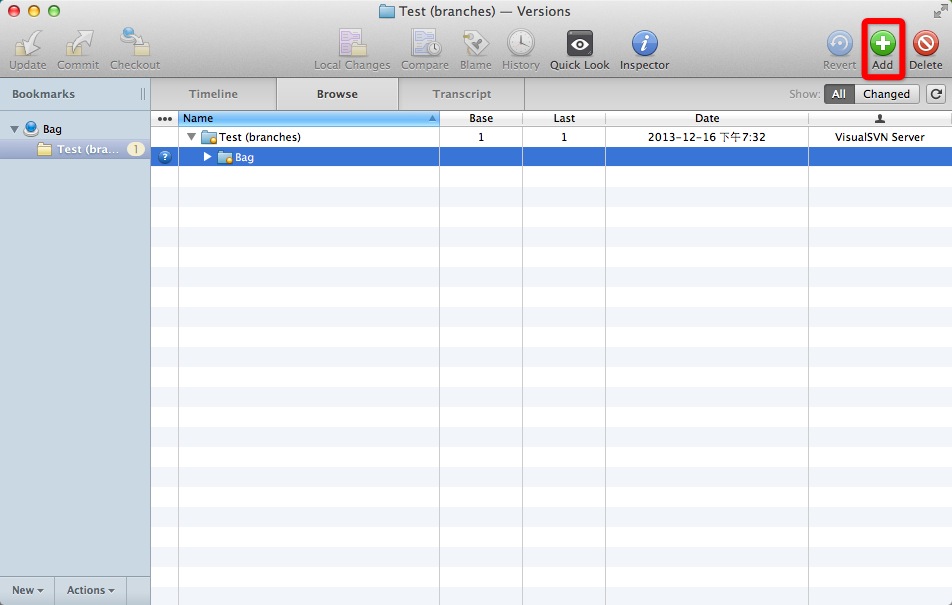
1. 首先我们下载Versions1.2.2 MAC版,下载之后进行安装,安装之后的主界面如下所示



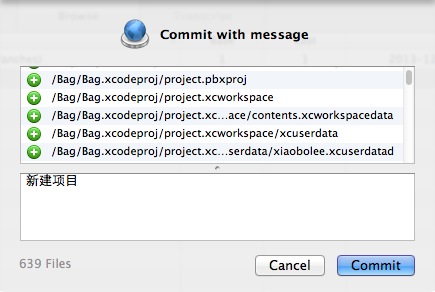
主界面中有两个按钮,New Repository Bookmark,这是一个新的代码仓库,第二个New Working Copy Bookmark这就是进行备份。在这里我们选择第一个New Repository Bookmark,然后进入如下界面

在name里面我们输入我们本地数据仓库的名字,在Location里面我们输入SVN服务器的地址<http://192.168.0.56:8080/svn/Test>(根据自己实际的IP地址进行修改,在你的服务器代码库上方有详细的地址),在UserName里输入test(我们服务器创建的测试用户名),Password里面输入你的密码,在这里我是123456,然后点击Create,这样我们就创建了一个数据仓库.

1. 下载服务器代码

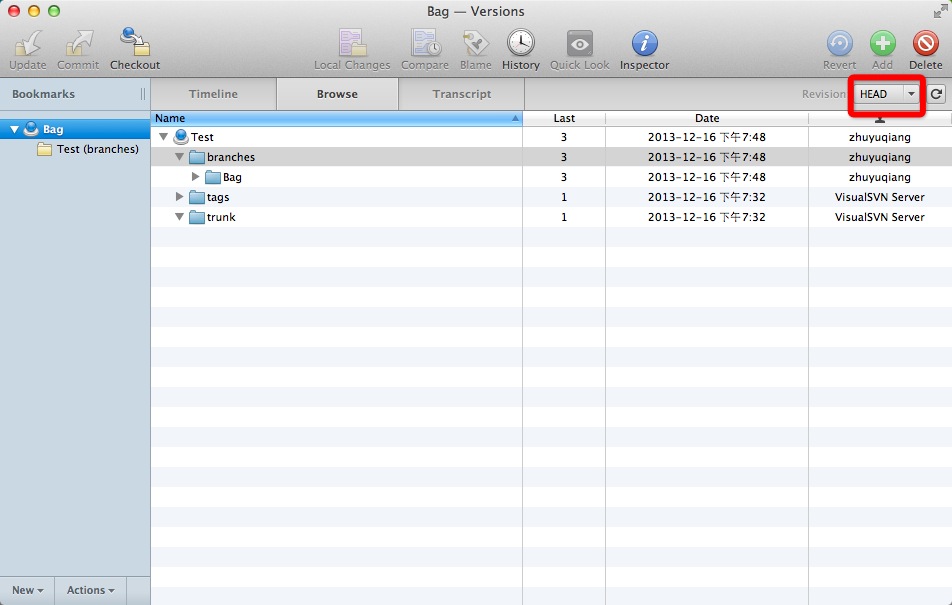
下面我们开始下载代码,拷贝到本地,指定一个工作区目录,点击CheckOut,然后就开始从服务器下载代码到本地.然后我们拷贝一个已有的工程放到刚才的目录下.再次回到客户端,可以发现里面有一个黄色的标志,表示我们修改了工作区的代码,选中然后点击Add,就可以保存发生的修改.

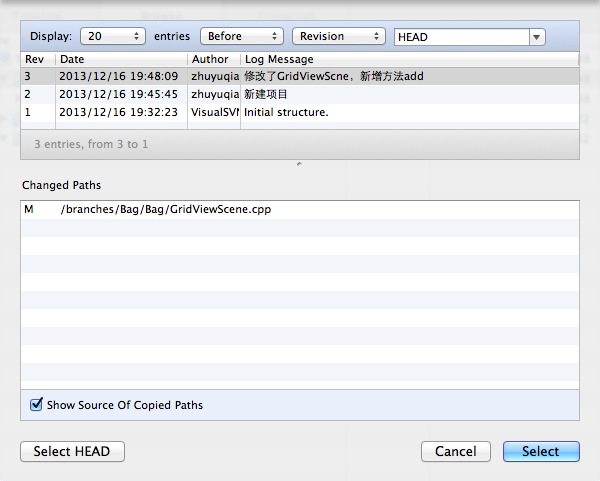
1. 提交代码

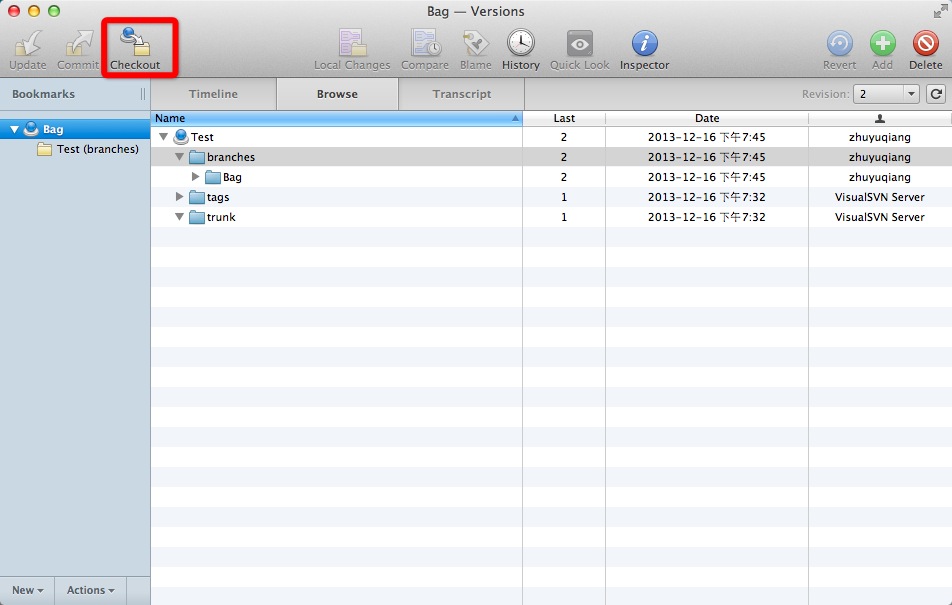
点击Commit按钮,然后会提交代码,提交的时候会出现如下提示,我们可以添加提交的注释,为了以后查看谁修改了代码,何时修改,然后点击Commit即可

1. 恢复代码

有的时候我们不小心把现在正在开发的代码搞崩溃了,我们需要恢复以前的版本,这时候我们需要一些操作.首先点击Revision，默认的选择Head,我们选择Browse otherRevision,会进入如下界面.





然后我们可以选择要恢复的版本进行恢复,点击Select即可,这时候回到主界面,然后重新CheckOut,保存即可,如下所示

以上就是我们的版本控制工具SVN的使用,当然还有很多其他的版本控制工具,像git,CVS等等,我们只需要了解一种即可,有兴趣的同学可以研究一下git,作为自己知识的扩充。