实验五：SVN的安装、配置和使用

|  |  |  |  |
| --- | --- | --- | --- |
| **学号：** | **班级：** | **姓名：** | **评分：** |

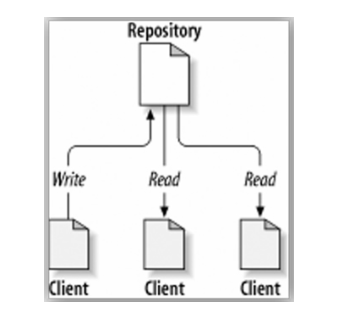
**一、实验目的**

掌握软件配置管理工具——SVN的安装、配置以及使用。

**二、实验内容**

TortoiseSVN是Subversion版本控制系统的一个免费开源客户端，可以跨时间管理文件和目录。文件保存在中央版本库，除了能记住文件和目录的每次修改以为，版本库非常像普通的文件服务器。我们可以将文件恢复到过去的版本，并且可以通过检查历史知道数据做了哪些修改，谁做的修改。

Subversion的核心就是repository, 翻译成“版本库”。它是存储工作拷贝文件集的仓库,对外提供一定的接口供SVN客户端（如TortoiseSVN）使用。对外接口可以是文件系统、SVN服务、apache插件提供的SVN服务,等等。一定要区别SVN版本库和工作拷贝。版本库以典型的文件和目录结构形式文件系统树来保存信息。任意数量的客户端连接到Subversion版本库,读取、修改这些文件。客户端通过写数据将信息分享给其他人,通过读取数据获取别人共享的信息,如图1所示。

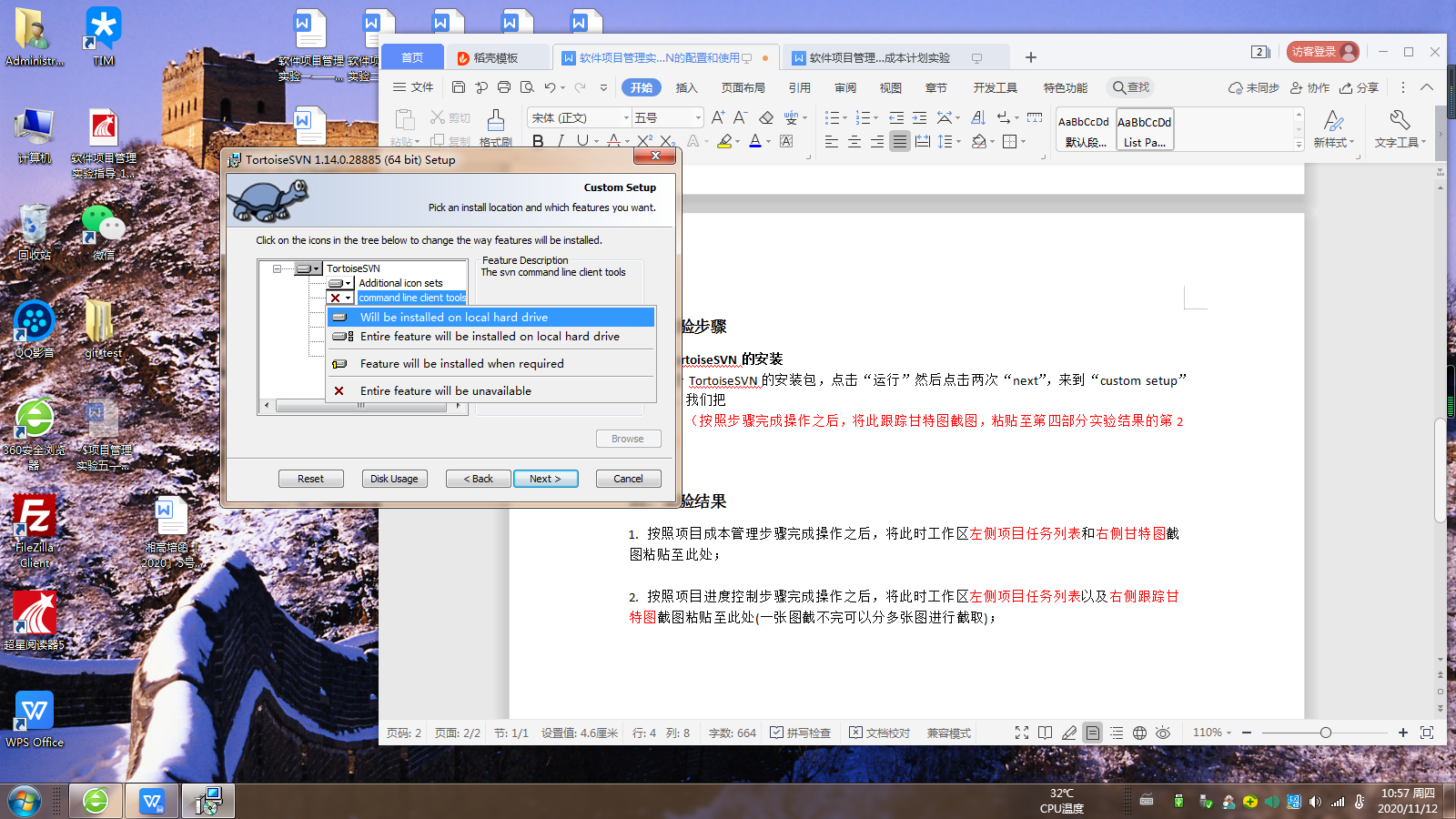
图1 SVN架构图

我们本次的实验内容主要为两个部分：

1. TortoiseSVN的安装和配置，并使用TortoiseSVN建立repository并通过两种方式(本地file形式和远程svn形式)checkout(签出)所建立的repository中的内容；
2. 对repository进行update以及commit等操作，进行文件的版本控制；
3. **实验步骤**

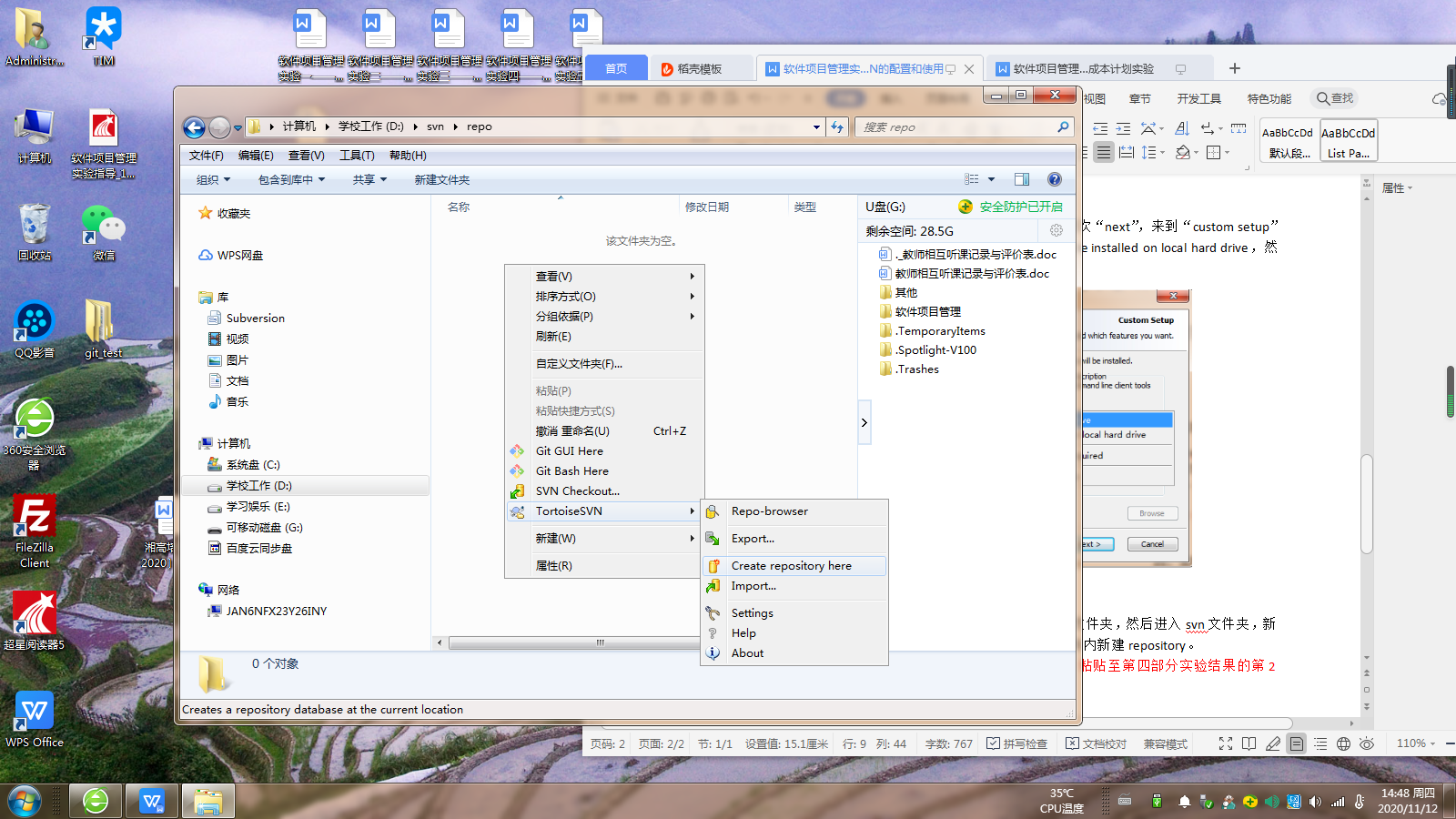
**1.TortoiseSVN的安装**

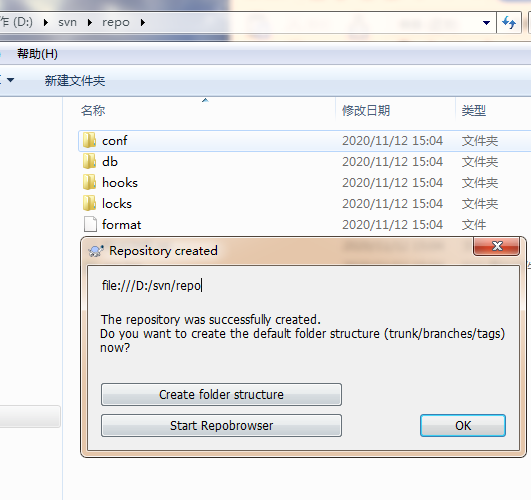
双击TortoiseSVN的安装包，点击“运行”然后点击两次“next”，来到“custom setup”界面时，我们把command line client tools选项中选择will be installed on local hard drive，然后点击“next”和“Install”完成安装。



**2.使用TortoiseSVN新建repository**

我们来到D盘的根目录，在D盘新建名称为svn的文件夹，然后进入svn文件夹，新建repo文件夹，然后进入repo文件夹，然后鼠标右键，在“TortoiseSVN”菜单中选择“Create repository here”，我们会在该文件夹内新建repository。

新建完repository之后，此时目录文件夹如下所示，单击“ok”确定即可：

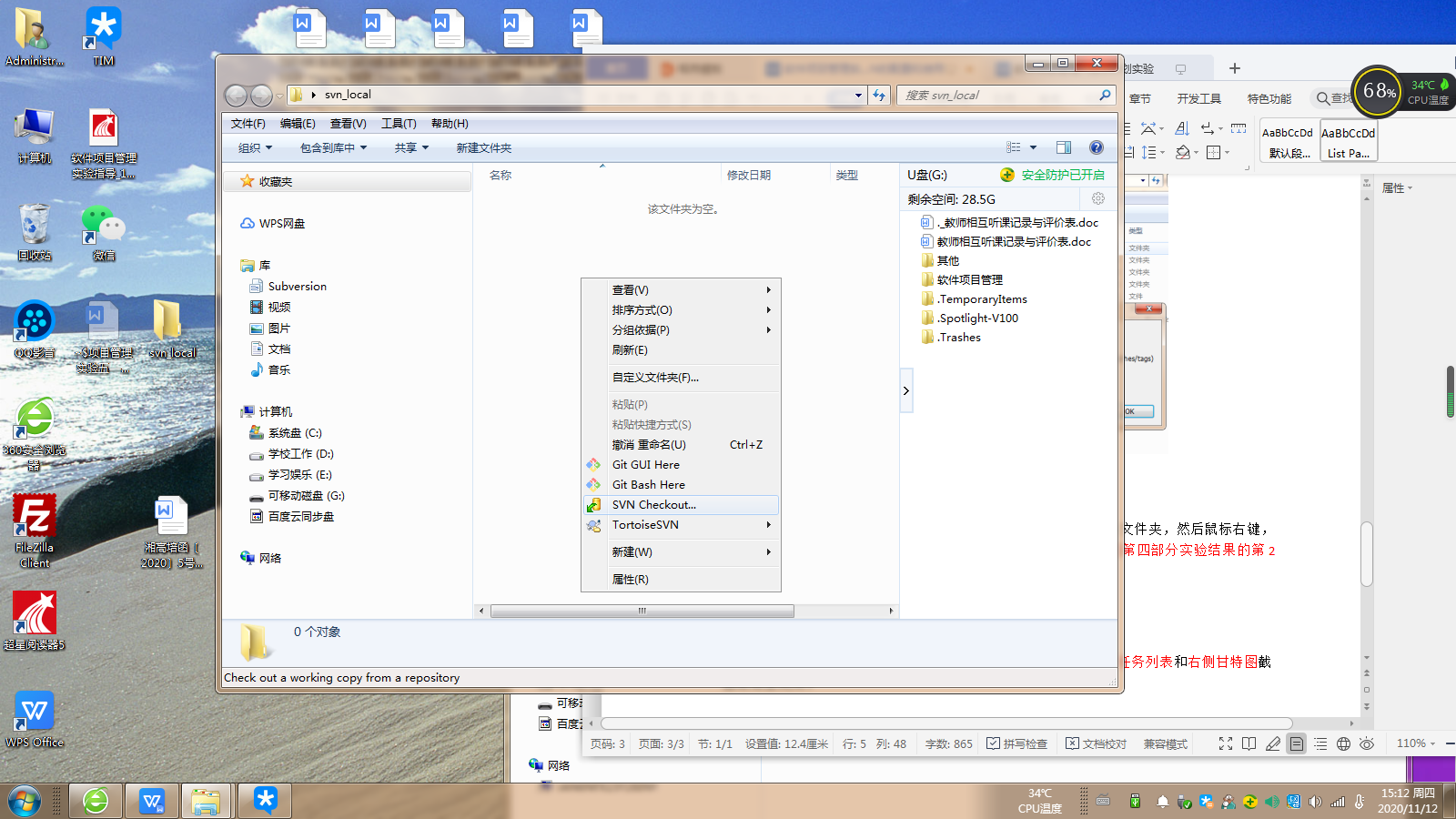


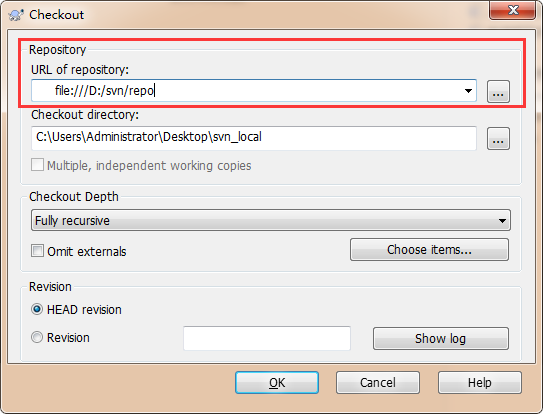
**3.签出repository**

签出repository即将repository中的内容拷贝一份至我们的工作目录，有两种方式可以对我们的repository进行签出。

1）使用本地仓库形式对repository进行签出

回到桌面，在桌面上新建svn\_local文件夹，并进入svn\_local文件夹，然后鼠标右键，选择“SVN Checkout”，如下所示：

在出现的窗口中，在URL of repository中填入下图中的地址：



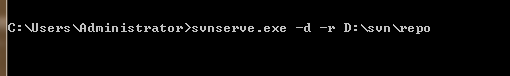
然后点击“ok”，此时，我们已经成功的将repository中的内容签出(checkout)，只不过由于我们的repository新建完成之后并没有写入内容，所以此时签出之后的文件夹里面只有“.svn”这一隐藏文件夹，该文件夹是svn的工作目录，一般不需要对其进行改动。



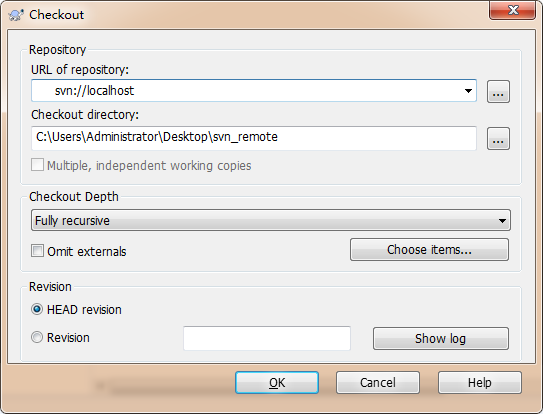
2）使用远程仓库形式对repository进行签出

TortoiseSVN功能之一就是能够帮助我们进协作开发，即所有的经过验证的客户端都能够对远程仓库repository中的代码进行更改，为了使用该功能，我们需要对TortoiseSVN进行一些配置。

首先我们打开cmd命令行工具，然后在命令行中输入“svnserve.exe -d -r D:\svn\repo”，回车之后可能会提示窗口，我们点击确定即可，注意不要关闭命令行窗口。接下来svnserve将会在端口3690等待请求，-d选项告诉其以守护进程方式运行，这样在手动终止之前不会退出，-r选项设置根目录位置，即我们在之前设置的repository的路径。

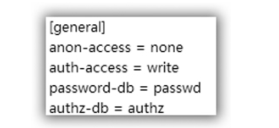


接下来，我们在桌面上新建svn\_remote文件夹，然后在文件夹中鼠标右键选择“SVN Checkout”，在窗口中，我们填入下图中的地址（注意与本地形式的区别），如图所示：



由于仓库服务器在本地，所以我们可以使用localhost，如果局域网中其他机器想要访问该repository，可将localhost改为本机ip地址。默认情况下，匿名用户只能进行Checkout操作（从远程repository下载代码）和update（更新本地工作目录代码使得本地工作目录代码和远程repository一致）操作，为了实现commit操作，我们还需要对repository进行一些配置，为svn加入用户验证。

首先打开版本库所在的文件夹,即D:\svn\repo,在repo文件夹下的conf文件夹里的svnserve.conf文件，在[ general]下面添加一-些内容,如图所示（文件中以#开头的行是被注释的行）：

其中：

anon-access = none表示匿名用户无法使用

auth-access = write表示通过用户认证的有写的权限

password-db = passwd表示用户密码配置文件，即为该目录下的passwd文件

authz-db = authz 表示用户组权限配置文件，即为该目录下的passwd文件

现在访问SVN就需要用户名和密码了，下面来配置一下用户名密码。首先打开passwd文件,在文件里的[ users]下面添加用户名和密码,格式为:用户名=密码,例如:

[ users ]

Tom=123

Jerry= 123

Jim= 123

这样就有了用户和密码，下面可以分配某个用户访问某个目录的权限。权限分配时,应遵守从根目录到子目录、从设置最广泛权限到最精细权限、从只读权限到读写权限的设置原则,即从根目录开始设置最广泛的访问权限,然后逐步设置下属子目录的访问权限。提示:目录的访问权限既可以分配给组，也可以分配给指定用户。

打开authz文件,在[groups]域里添加组,格式为:组名=用户名,用户名...。现在创建一个developer组,将Tom和Jim都加进去,我们让Tom和Jim作为开发者;而Jerry作为设计者,加入到design组中。也就是:

[ groups]

developer= Tom,Jim

design= Jerry

然后在下面添加权限控制,我们让开发者developer这个组的人可以访问所有目录(拥有读写权限);而设计者只有只读权限。

[/] 井代表根目录

@developer=rw #r 代表读，w代表写,rw代表有读和写的权限

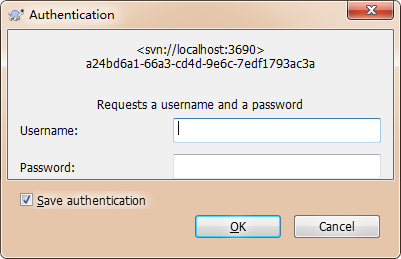
@design= r

在此不再做过于详细的权限控制,但是权限设置是非常必要的,可以参照以上说明自己设计。

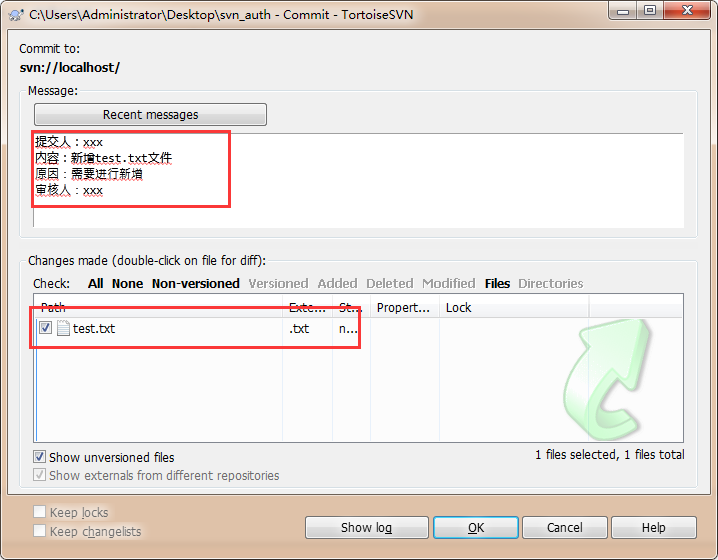
（此处要求同学们设置一个用户名与密码均为自己学号的账户，并将该账户添加至developer的组中，然后将有自己学号部分的passwd文件部分和authz文件部分进行截图，粘贴至第四部分实验结果中的第1题。）

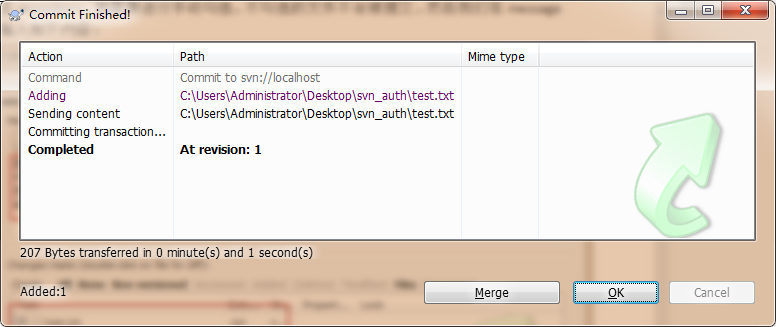
**3.commit操作**

在完成上述操作之后，我们回到桌面，新建svn\_auth文件夹，我们进入到该文件夹，然后鼠标右键，选择“SVN Checkout”，在地址中填入“svn://localhost”，然后点击确定，此时会要求我们输入账户和密码（前一步操作中让大家设置了自己的学号作为用户名和密码），输入自己的学号作为账户名和密码，然后点击ok即可（默认勾选save authentication，下次checkout、update以及commit时就不需要再次输入了）。



当checkout操作完成之后，我们在该文件夹中，新建一个“test.txt”文件，然后鼠标右键，选择“SVN commit”，该操作可以将我们新增或是修改的内容，上传至repository，此时会出现如下对话窗口，svn会自动将我们修改过或者新增的文件列出来，如果这些文件需要被提交至repository，就需要进行手动勾选，不勾选的文件不会被提交，然后我们在message窗口中输入如下内容：

提交人和审核人填写自己的名字即可，然后点击“ok”，即可将内容上传至repository。

（提交完成之后，我们在文件中选择鼠标右键，然后选择“TortoiseSVN”，然后选择“Show log”，将此时的Show log窗口截图，提粘贴至第四部分实验结果中第2题。）

**四、实验结果**

1. 按照步骤将有自己学号部分的passwd文件部分和authz文件部分进行截图，粘贴至此处；

1. 按照步骤将show log进行截图，粘贴至此处；