

# SVN

<b>SVN .....</b>	<b>3</b>
一、 什么是 SVN .....	3
二、 SVN 的特性 .....	3
1 数据存储 .....	3
2 运行方式 .....	3
3 优缺点简述 .....	3
3.1 速度 .....	3
3.2 标志&分支 .....	3
3.3 文件类型 .....	3
三、 安装 SVN 服务器 .....	4
1 安装环境 .....	4
2 使用账户 .....	4
3 安装 Apache HTTP Server .....	4
3.1 前置准备（可选） .....	4
3.2 安装 Apache HTTP Server .....	4
3.3 修改配置 .....	4
3.4 修改防火墙配置 .....	4
3.5 启动 Apache HTTP Server 服务 .....	5
3.6 测试访问 .....	5
4 安装 SVN 服务 .....	5
4.1 停止 Apache HTTP Server 服务 .....	5
4.2 安装 svn 模块 .....	5
4.3 检查 svn 模块安装结果 .....	5
4.4 为 svn 仓库创建主目录 .....	6
4.5 修改 svn 配置信息 .....	6
4.6 创建密码文件和授权文件 .....	7
5 安装 MySQL 数据库 .....	7
5.1 检查当前系统中是否安装了 MySQL .....	7
5.2 安装 MySQL .....	7
5.3 启动 MySQL 服务 .....	7
5.4 检查 MySQL 服务状态 .....	7
5.5 修改防火墙配置 .....	7
5.6 设置 MySQL 数据库 root 用户初始密码 .....	7
5.7 为 MySQL 数据库 root 用户授权 .....	7
5.7.1 连接 MySQL 数据库 .....	8
5.7.2 授权 .....	8
5.7.3 查看授权信息 .....	8
6 安装 svnadmin .....	8
6.1 简介 .....	8
6.2 svnadmin 的优点 .....	8

6.3	安装 <i>Tomcat</i> .....	9
6.3.1	解压缩 .....	9
6.3.2	重命名目录 .....	9
6.3.3	修改 <i>tomcat</i> 配置（可选） .....	9
6.3.4	修改防火墙配置 .....	9
6.4	部署 <i>svnadmin</i> 应用 .....	9
6.5	修改 <i>svnadmin</i> 应用配置 .....	9
6.6	创建 <i>database</i> &导入 <i>SQL</i> 脚本 .....	10
6.6.1	连接 <i>MySQL</i> .....	10
6.6.2	创建 <i>database</i> .....	10
6.6.3	使用 <i>database</i> .....	10
6.6.4	执行脚本文件 .....	10
6.7	启动 <i>httpd</i> 服务 .....	10
6.8	启动 <i>Tomcat</i> .....	10
6.9	测试访问 <i>svnadmin</i> 应用 .....	11
7	管理版本仓库 .....	11
7.1	初始化管理账户 .....	11
7.2	创建版本库 .....	11
7.3	用户权限问题 .....	12
7.3.1	为 <i>SVN</i> 仓库分配用户 .....	12
7.3.2	测试用户访问权限 .....	14
7.3.3	关闭 <i>seLinux</i> .....	15
7.3.4	结果显示 .....	16
7.4	仓库授权 .....	16
7.4.1	创建用户 .....	16
7.4.2	用户组管理 .....	17
7.4.3	用户组授权 .....	18
7.4.4	授权文件的变化 .....	19

## SVN

### 一、 什么是 SVN

SVN 是 *Subversion* 的简称，是一个开放源代码的版本控制系统，相较于 *RCS*、*CVS*，它采用了分支管理系统，它的设计目标就是取代 *CVS*。互联网上很多版本控制服务已从 *CVS* 迁移到 *Subversion*。说得简单一点 SVN 就是用于多个人共同开发同一个项目，共用资源的目的。

### 二、 SVN 的特性

#### 1 数据存储

svn 存储版本数据也有 2 种方式：*BDB*(一种事务安全型表类型)和 *FSFS*(一种不需要数据库的存储系统)。因为 *BDB* 方式在服务器中断时，有可能锁住数据，所以还是 *FSFS* 方式更安全一点。

#### 2 运行方式

svn 服务器有 2 种运行方式：独立服务器和借助 *apache* 运行。两种方式各有利弊，用户可以自行选择。

#### 3 优缺点简述

##### 3.1 速度

*CVS* 比较慢。

整体而言,由于架构实现的不同, *SVN* 的确比 *CVS* 快很多。在网络上它只传输很少的信息并支持更多的离线模式的功能。但这也是有代价的。速度的代价就是巨大的存储（完全备份所有的工作文件）。

##### 3.2 标志&分支

*SVN* 采用标志和分支而抛弃了其他三件东西，实际上这意味着他们把这个概念替换为在档案库内部复制文件或目录以便保存日志。这样一来，无论标志创建还是分支创建都只是仓库内部的文件复制了。对分支而言：分支不过是在仓库内部的一个单独的目录而已了，不像早期还有些什么交错。对标志而言：已经不能对代码加标志了。在某种程度上说，*SVN* 全文编号补足了这个缺陷，*SVN* 里整个仓库都有版本号，但不是针对单个文件。

##### 3.3 文件类型

*SVN* 会关心所有的文件类型，不需要你来手工操作。

*CVS* 最初是为文本文件存储而设计的。因此其他文件类型（二进制，统一码）文件的支

持几乎没有，如需要的话则要有其他信息，并且客户端服务器端都要调整。

### 三、 安装 SVN 服务器

独立服务器安装方式简单，由同学自学安装。  
本课程安装借助 Apache 运行的 SVN 服务器。

#### 1 安装环境

Linux CentOS。

#### 2 使用账户

root 根用户。此用户为系统根用户，拥有最高权限。SVN 服务需要频繁访问系统资源，使用 root 用户可以避免一些不必要的麻烦。且 SVN 服务器通常为独立服务器，使用 root 用户可以更好的管理服务器资源。

#### 3 安装 Apache HTTP Server

##### 3.1 前置准备（可选）

在安装前，可以更新当前操作系统，减少漏洞。

更新命令：**yum update**

更新系统后，需要重新启动操作系统。

重启命令：**reboot**

##### 3.2 安装 Apache HTTP Server

命令：**yum install httpd httpd-devel -y**

##### 3.3 修改配置

命令：**vi /etc/httpd/conf/httpd.conf**

找到 **ServerName** 配置位置（vi 编辑器命令模式下输入：**/ServerName**），修改内容如下：

```
ServerName localhost:80
```

##### 3.4 修改防火墙配置

开放 80 端口访问权限。

命令：**vi /etc/sysconfig/iptables**

增加下述内容：

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
```

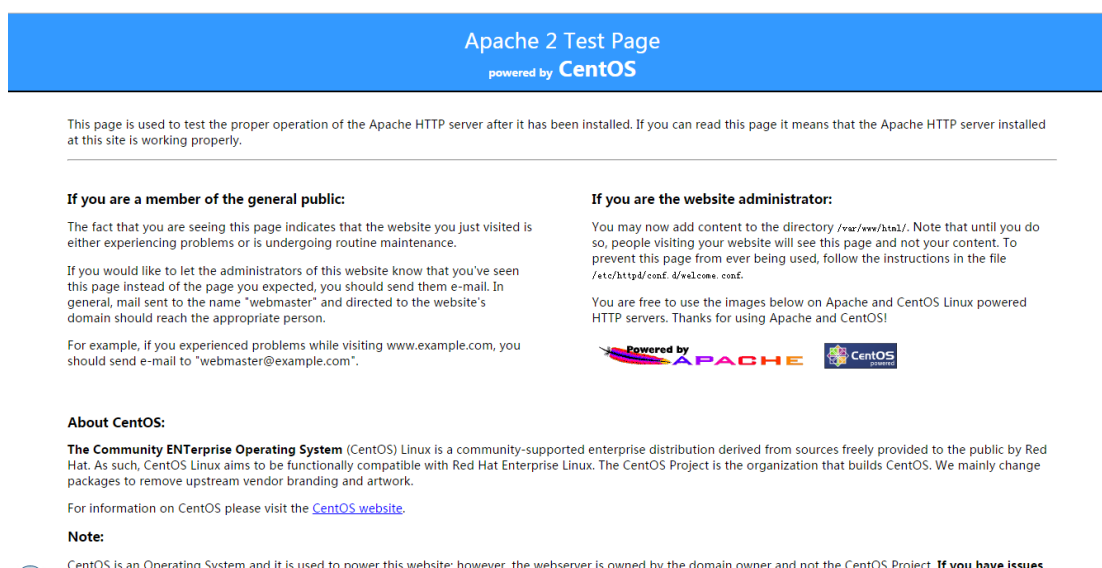
修改后，重启防火墙服务，命令：**`service iptables restart`**

## 3.5 启动 Apache HTTP Server 服务

命令：**`service httpd start`**

## 3.6 测试访问

使用浏览器访问对应服务。本课程使用 Linux 虚拟机 IP 为：192.168.120.159。访问地址为：<http://192.168.120.159/>。访问结果如下：



## 4 安装 SVN 服务

### 4.1 停止 Apache HTTP Server 服务

命令：**`service httpd stop`**

### 4.2 安装 svn 模块

命令：**`yum install mod_dav_svn subversion`**

### 4.3 检查 svn 模块安装结果

当 svn 模块安装成功后，在 `/etc/httpd/modules/` 目录中会有对应信息产生，可使用下述命令查看：

**`ls /etc/httpd/modules | grep svn`**

结果如下：

```
mod_auth_svn.so
mod_dav_svn.so
```

也可以使用 svn 模块命令查看，命令为：**`svn --version`**

结果如下：

```
svn, version 1.6.11 (r934486)
  compiled Aug 17 2015, 08:37:43

Copyright (C) 2000-2009 CollabNet.
Subversion is open source software, see http://subversion.tigris.org/
This product includes software developed by CollabNet (http://www.Collab.Net/).

The following repository access (RA) modules are available:

* ra_neon : Module for accessing a repository via WebDAV protocol using Neon.
  - handles 'http' scheme
  - handles 'https' scheme
* ra_svn : Module for accessing a repository using the svn network protocol.
  - with Cyrus SASL authentication
  - handles 'svn' scheme
* ra_local : Module for accessing a repository on local disk.
  - handles 'file' scheme
```

## 4.4 为 svn 仓库创建主目录

本课程将 svn 仓库主目录保存在系统根目录下，可根据具体情况修改。

**mkdir /svn/**

## 4.5 修改 svn 配置信息

svn 配置文件保存在目录/etc/httpd/conf.d/目录中，配置文件命名为：subversion.conf。  
此配置文件是在安装 mod\_dav\_svn 时自动创建的。

命令：**vi /etc/httpd/conf.d/subversion.conf**

在配置文件中增加下述内容：

```
#Include /svn/httpd.conf
<Location /svn/> # 本地仓库目录位置。4.4 节点创建的位置
  DAV svn # 模型简述
  SVNListParentPath on # 开启上级目录列表能力
  SVNParentPath /svn # 上级目录列表根目录。4.4 节点创建的位置
  AuthType Basic # 授权类型，
  AuthName "Subversion repositories" # 拥有者名称
  AuthUserFile /svn/passwd.http # 拥有者用户文件命名
  AuthzSVNAccessFile /svn/authz # 用户授权文件命名
  Require valid-user
</Location>
RedirectMatch ^(/svn)$ $1/
```

## 4.6 创建密码文件和授权文件

命令：

```
touch /svn/passwd.http
```

```
touch /svn/authz
```

## 5 安装 MySQL 数据库

此数据库专为 SVN 服务器提供数据存储。与业务系统数据库分离。

### 5.1 检查当前系统中是否安装了 MySQL

命令：`rpm -qa | grep mysql`

### 5.2 安装 MySQL

命令：`yum install mysql-server mysql mysql-devel`

### 5.3 启动 MySQL 服务

命令：`service mysqld start`

在安装权限 MySQL 数据库后，第一次启动需要初始化数据库基础信息。

### 5.4 检查 MySQL 服务状态

命令：`service mysqld status`

### 5.5 修改防火墙配置

开放 3306 端口访问权限。

命令：`vi /etc/sysconfig/iptables`

增加下述内容：

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 3306 -j ACCEPT
```

修改后，重启防火墙服务，命令：`service iptables restart`

### 5.6 设置 MySQL 数据库 root 用户初始密码

命令：`mysqladmin -uroot password 'yourPassword'`

### 5.7 为 MySQL 数据库 root 用户授权

在默认情况下，MySQL 数据库中的 root 用户只能本地访问。需要为此用户授予远程访问权限，否则无法在远程或其他应用中使用 root 用户访问 MySQL 数据库。

### 5.7.1 连接 MySQL 数据库

命令：`mysql -uroot -p`

### 5.7.2 授权

MySQL 中的命令：

```
GRANT ALL PRIVILEGES ON *.* TO 'root'@'%' IDENTIFIED BY 'yourPassword' WITH GRANT OPTION;  
FLUSH PRIVILEGES;
```

### 5.7.3 查看授权信息

MySQL 中的命令：

```
select user, host from mysql.user;
```

其中 `user` 为 MySQL 数据库用户名，`host` 为可访问的地址（%代表所有位置都可访问）。

## 6 安装 svnadmin

### 6.1 简介

`Svn Admin` 是一个 `Java` 开发的管理 `Svn` 服务器的项目用户的 `web` 应用。安装好 `Svn` 服务器端好，把 `Svn Admin` 部署好，就可以通过 `web` 浏览器管理 `Svn` 的项目，管理项目的用户，管理项目的权限。使得管理配置 `Svn` 简便，再也不需要每次都到服务器手工修改配置文件。

本课程中使用的版本为：**3.0.5**

`svnadmin` 资源在 `googlecode` 平台中，不易访问下载，课程附件中附有资源。

网址为：<https://code.google.com/p/svnadmin>

### 6.2 svnadmin 的优点

多数据库：`Svn` 项目配置数据保存在数据库，支持所有数据库（默认 `MySQL/Oracle/SQL Server`）。

多操作系统：支持 `Window/Linux` 等操作系统。

权限控制：管理员可以随意分配权限、项目管理员可以管理项目成员、成员只能查看和修改自己的密码。

支持多项目、多用户、多用户组 `Group`（默认带有“项目管理组”、“项目开发组”、“项目测试组”）。

安全：密码加密保存。

多协议：支持 `svn` 协议和 `http` 协议(从 2.0 开始支持 `Apache` 服务器单库方式，从 3.0 开始支持 `Apache` 多库方式)

仓库浏览(从 3.0.2 开始)

多语言(从 3.0.2 开始)



## 6.3 安装 Tomcat

`svnadmin` 是使用 `java` 编写的一个 `WAR` 应用，需要提供一个 `Java WEB` 中间件才能正常使用，本课程使用 `Tomcat7` 作为部署容器。本课程中 `tomcat` 安装位置为：`/root/upload/`目录。  
`Tomcat` 安装必须依赖 `JDK`，在 `Maven` 课程已安装过，此处省略。

### 6.3.1 解压缩

命令：`tar -zxvf apache-tomcat-7.0.68.tar.gz`

### 6.3.2 重命名目录

命令：`mv apache-tomcat-7.0.68 tomcat-svnadmin`

### 6.3.3 修改 tomcat 配置（可选）

命令：`vi /root/upload/tomcat-svnadmin/conf/server.xml`

修改内容如下：

```
<!-- 标红部分为可选修改内容 -->
<Server port="8005" shutdown="SHUTDOWN">
<Connector port="8080" protocol="HTTP/1.1"
            connectionTimeout="20000"
            redirectPort="8443" URIEncoding="UTF-8"/>
```

### 6.3.4 修改防火墙配置

开放 `8080` 端口访问权限。

命令：`vi /etc/sysconfig/iptables`

增加下述内容：

```
-A INPUT -m state --state NEW -m tcp -p tcp --dport 8080 -j ACCEPT
```

修改后，重启防火墙服务，命令：`service iptables restart`

## 6.4 部署 svnadmin 应用

将 `svnadmin` 应用上传到 `Tomcat/webapps/`目录中，并解压缩。`svnadmin-3.0.5.zip` 文件中包含 `db` 目录，`svnadmin.war` 应用。`db` 目录中保存的是 `svnadmin` 需要使用的数据库脚本文件。

解压缩命令：`unzip svnadmin.war -d svnadmin`

## 6.5 修改 svnadmin 应用配置

命令：`vi /root/upload/tomcat7/webapps/svnadmin/WEB-INF/jdbc.properties`

修改内容如下：

```
db=MySQL
#MySQL
MySQL.jdbc.driver=com.mysql.jdbc.Driver
MySQL.jdbc.url=jdbc:mysql://127.0.0.1:3306/svnadmin?characterEncoding=utf-8
MySQL.jdbc.username=root
MySQL.jdbc.password=root
```

## 6.6 创建 database&导入 SQL 脚本

### 6.6.1 连接 MySQL

命令：`mysql -uroot -p`

### 6.6.2 创建 database

MySQL 中的命令：`create database svnadmin default charset utf8 collate utf8_general_ci;`

这里的 `database` 命名需要和应用中配置文件 `jdbc.properties` 内的 `url` 对应。

### 6.6.3 使用 database

MySQL 中的命令：`use svnadmin`

### 6.6.4 执行脚本文件

本课程中 `db` 目录保存在 `/root/upload/` 目录中。

MySQL 中的命令：

```
source /root/upload/db/mysql5.sql
source /root/upload/db/lang/en.sql
```

## 6.7 启动 httpd 服务

命令：`service httpd start`

## 6.8 启动 Tomcat

命令：

```
/root/upload/tomcat7/bin/startup.sh & tail -f /root/upload/tomcat7/log/catalina.out
```

如果 `Tomcat` 是第一次使用，`catalina.out` 日志文件不存在，无法直接输出日志信息。

## 6.9 测试访问 svnadmin 应用

浏览器访问：<http://192.168.120.159:8080/svnadmin>

结果如下：

欢迎使用SVN ADMIN,第一次使用请设置管理员帐号和密码.

帐号

密码

## 7 管理版本仓库

### 7.1 初始化管理账户

欢迎使用SVN ADMIN,第一次使用请设置管理员帐号和密码.

帐号

密码

本课程使用 `admin` 用户，密码为 `root`。请同学自定义用户。

zh\_CN ▼

**SVN ADMIN** V 3.0.5

用户 项目 语言 **admin** 退出

---

**项目管理**

项目	<input type="text"/>	*	类型	svn ▼	*
路径	<input type="text"/>				
URL	<input type="text"/>				
描述	<input type="text"/>				
<input type="button" value="提交"/>					

初始化用户后进入的界面

NO.	项目	路径	URL	类型	描述	设置用户	设置用户组	设置权限	删除
-----	----	----	-----	----	----	------	-------	------	----

### 7.2 创建版本库

创建的版本仓库任意信息包含中文，必须要求，**数据库的对应 `database` 字符集支持中文，数据库导入的 `sql` 文件必须有 `en.sql` 数据，`tomcat` 中的 `server.xml` 配置文件中必须增加 `URIEncoding="UTF-8"` 信息。**

zh\_CN ▼

**SVN ADMIN** V 3.0.5

用户 项目 语言 admin 退出

项目管理

库名自定义

类型为http协议多库

项目  \*

类型  \*

路径  \*

URL  \*

描述  \*

URL: http协议访问SVN所在主机, 后续地址为库路径。模拟WEB寻址方式, 从系统根路径开始寻址。

描述自定义

NO.	项目	路径	URL	类型	描述	设置用户	设置用户组	设置权限	删除

项目管理

项目  \*

类型  \*

路径  \*

URL  \*

描述  \*

创建版本库成功后结果

NO.	项目	路径	URL	类型	描述	设置用户	设置用户组	设置权限	删除
1	<a href="#">sxt_svn</a>	/svn/sxt_svn	<a href="http://192.168.120.159/svn/sxt_svn">http://192.168.120.159/svn/sxt_svn</a>	http-mutil	架构课程-工程构建管理-SVN		<a href="#">设置用户组</a>	<a href="#">设置权限</a>	<a href="#">删除</a>

在版本库创建成功后, 在 Linux 系统中, 会有一个目录与之对应。目录为: `/svn/sxt_svn`。

在之前创建的 `/svn` 目录中会有新文件 `httpd.conf` 创建。其中内容和 `/etc/httpd/conf.d/subversion.conf` 一致。

`/svn/authz` 授权文件中会有如下内容:

```
[aliases]
[groups]
sxt_svn_developer=
sxt_svn_manager=
sxt_svn_tester=

[sxt_svn:/]
@sxt_svn_manager=rw
```

## 7.3 用户权限问题

### 7.3.1 为 SVN 仓库分配用户

NO.	项目	路径	URL	类型	描述	设置用户	设置用户组	设置权限	删除
1	<a href="#">sxt_svn</a>	/svn/sxt_svn	<a href="http://192.168.120.159/svn/sxt_svn">http://192.168.120.159/svn/sxt_svn</a>	http-mutil	架构课程-工程构建管理-SVN		<a href="#">设置用户组</a>	<a href="#">设置权限</a>	<a href="#">删除</a>

项目	sxt_svn	用户组	<input type="text"/>	*	描述	<input type="text"/>	提交
----	---------	-----	----------------------	---	----	----------------------	----

NO.	项目	用户组	描述	设置用户	删除
1	sxt_svn	<a href="#">developer</a>	developer	<a href="#">设置用户</a>	<a href="#">删除</a>
2	sxt_svn	manager	manager	<a href="#">设置用户</a>	
3	sxt_svn	<a href="#">tester</a>	tester	<a href="#">设置用户</a>	<a href="#">删除</a>

维护管理组用户，其他组用户由同学自行测试

项目管理(sxt\_svn) --> 用户组管理(manager)-->项目组用户管理

NO.	项目	用户组	帐号	删除
-----	----	-----	----	----

项目管理(sxt\_svn) --> 用户组管理(manager)-->项目组用户管理

NO.	项目	用户组	帐号	删除
-----	----	-----	----	----

## 项目管理(sxt\_svn) --> 用户组管理(manager)-->项目组用户管理

NO.	项目	用户组	帐号	删除
1	sxt_svn	manager	admin	<a href="#">删除</a>

### 7.3.2 测试用户访问权限

在创建好版本仓库并分配好用户后，在访问版本仓库的时候会出现认证失败错误。

NO.	项目	路径	URL	类型	描述	设置用户	设置用户组	设置权限	删除
1	sxt_svn	/svn/sxt_svn	http://192.168.120.159/svn/sxt_svn	http-mutil	架构课程-工程构建管理-SVN		设置用户组	设置权限	删除

#### 项目管理(sxt\_svn) --> 权限管理

```
[ERROR] 2018-04-15 23:23:33,020 org.svnadmin.service.RepTreeNodeService:85
认证失败
java.lang.RuntimeException: 认证失败
    at org.svnadmin.service.RepositoryService.getDir(RepositoryService.java:187)
    at org.svnadmin.service.RepTreeNodeService.getTreeNodes(RepTreeNodeService.java:60)
    at org.svnadmin.service.RepTreeNodeService.getHTML(RepTreeNodeService.java:23)
    at org.svnadmin.service.RepTreeNodeService.parseTree(RepTreeNodeService.java:72)
    at org.svnadmin.service.RepTreeNodeService.getHTML(RepTreeNodeService.java:51)
    at org.svnadmin.service.DefaultTreeService.execute(DefaultTreeService.java:34)
    at org.svnadmin.servlet.AjaxServlet.execute(AjaxServlet.java:81)
    at org.svnadmin.servlet.ServletSupport.doPost(ServletSupport.java:44)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:650)
    at javax.servlet.http.HttpServlet.service(HttpServlet.java:731)
    at org.apache.catalina.core.ApplicationFilterChain.internalDoFilter(ApplicationFilterChain.java:303)
    at org.apache.catalina.core.ApplicationFilterChain.doFilter(ApplicationFilterChain.java:208)
    at org.apache.tomcat.websocket.server.WsFilter.doFilter(WsFilter.java:52)
```

后台Tomcat抛出的异常信息

上述错误出现的原因是：/svn 目录中的仓库对应目录信息为 root 用户创建的，其他用户是没有访问权限的。而我们在访问 SVN 仓库的时候，是需要通过 Apache 的 HTTP Server 来访问的。这个时候，对 SVN 仓库目录的访问用户不再是 root，而是 apache。此用户的信息可以通过 Apache HTTP Server 的配置文件查看 (/etc/httpd/conf/httpd.conf)。具体如下：

```
#
# If you wish httpd to run as a different user or group, you must run
# httpd as root initially and it will switch.
#
# User/Group: The name (or #number) of the user/group to run httpd as.
# . On SCO (ODT 3) use "User nouser" and "Group nogroup".
# . On HP-UX you may not be able to use shared memory as nobody, and the
# suggested workaround is to create a user www and use that user.
# NOTE that some kernels refuse to setgid(Group) or semctl(IPC_SET)
# when the value of (unsigned)Group is above 60000;
# don't use Group #-1 on these systems!
#
User apache
Group apache
```

也就是说，此时访问 SVN 仓库目录的用户是 *apache* 组中的 *apache* 用户。需要将仓库目录的拥有者变更为 *apache* 用户。具体命令如下：

**chown -R apache.apache sxt\_svn**

命令解释：chown -R 组.用户 要修改拥有者的目录名。

变更拥有者前：

```
-rw-r--r--. 1 root root 105 Apr 15 23:05 authz
-rw-r--r--. 1 root root 258 Apr 15 23:05 httpd.conf
-rw-r--r--. 1 root root 0 Apr 15 23:05 passwd.http
drwxr-xr-x. 6 root root 4096 Apr 15 23:05 sxt_svn
```

变更拥有者后：

```
-rw-r--r--. 1 root root 110 Apr 15 23:22 authz
-rw-r--r--. 1 root root 258 Apr 15 23:22 httpd.conf
-rw-r--r--. 1 root root 40 Apr 15 23:22 passwd.http
drwxr-xr-x. 6 apache apache 4096 Apr 15 23:05 sxt_svn
```

**每次新增 SVN 仓库（项目）时，都需要修改目录拥有者。**

### 7.3.3 关闭 seLinux

在修改 SVN 仓库目录拥有者后，仍旧没有对仓库的访问权限。此时就是 *seLinux* 的问题了。我们通过配置文件来管理 *seLinux* 即可。

命令：vi /etc/selinux/config

修改内容如下：

```
# This file controls the state of SELinux on the system.
# SELINUX= can take one of these three values:
#     enforcing - SELinux security policy is enforced.
#     permissive - SELinux prints warnings instead of enforcing.
```

```
# disabled - No SELinux policy is loaded.
# SELINUX=enforcing # 修改前
SELINUX= disabled # 修改后
```

重启 Linux 即可。命令为：**reboot**

此操作完成一次即可。

### 7.3.4 结果显示

项目管理(sxt\_svn) --> 权限管理



NO.	项目	资源	用户组/帐号	权限	删除
1	sxt_svn	[sxt_svn/]manager		可读可写	<a href="#">删除</a>

## 7.4 仓库授权

当权限问题解决后，开始为我们的版本仓库提供用户和授权管理。

### 7.4.1 创建用户

SVN ADMIN v 3.0.5

[用户](#) [项目](#) [语言](#) [admin](#) [退出](#)

项目管理



NO.	项目	路径	URL	类型	描述	设置用户	设置用户组	设置权限	删除
1	sxt_svn	/svn/sxt_svn	http://192.168.120.159/svn/sxt_svn	http-mutil	架构课程-工程构建管理-SVN	设置用户	设置用户组	设置权限	删除



## 用户管理

帐号  \* 密码  角色  提交

NO.	帐号	密码	角色	删除	查看权限
1	<a href="#">admin</a>	cm9vdA**	admin	<a href="#">删除</a>	<a href="#">查看权限</a>

注意：如果创建的用户不是管理员用户，角色不做修改

## 查看权限

NO.	项目	描述	帐号	资源	权限
1	sxt_svn	架构课程-工程构建管理-SVN	admin	[sxt_svn:/]	可读可写

## 用户管理

帐号  \* 密码  角色  提交

NO.	帐号	密码	角色	删除	查看权限
1	<a href="#">admin</a>	cm9vdA**	admin	<a href="#">删除</a>	<a href="#">查看权限</a>
2	<a href="#">test_svn</a>	cm9vdA**		<a href="#">删除</a>	<a href="#">查看权限</a>

## 查看权限

NO.	项目	描述	帐号	资源	权限
-----	----	----	----	----	----

## 7.4.2 用户组管理

NO.	项目	路径	URL	类型	描述	设置用户	设置用户组	设置权限	删除
1	<a href="#">sxt_svn</a>	/svn/sxt_svn	<a href="http://192.168.120.159/svn/sxt_svn">http://192.168.120.159/svn/sxt_svn</a>	http-mutil	架构课程-工程构建管理-SVN		<a href="#">设置用户组</a>	<a href="#">设置权限</a>	<a href="#">删除</a>

## 项目管理(sxt\_svn) --> 用户组管理

项目  用户组  \* 描述  提交

NO.	项目	用户组	描述	设置用户	删除
1	sxt_svn	<a href="#">developer</a>	developer	<a href="#">设置用户</a>	<a href="#">删除</a>
2	sxt_svn	<a href="#">manager</a>	manager	<a href="#">设置用户</a>	
3	sxt_svn	<a href="#">tester</a>	tester	<a href="#">设置用户</a>	<a href="#">删除</a>

选择要管理的用户组

项目管理(sxt\_svn) --> 用户组管理(developer)-->项目组用户管理

NO.	项目	用户组	帐号	删除
1	sxt_svn	developer	test_svn	<a href="#">删除</a>

### 7.4.3 用户组授权

NO.	项目	路径	URL	类型	描述	设置用户	设置用户组	设置权限	删除
1	<a href="#">sxt_svn</a>	/svn/sxt_svn	<a href="http://192.168.120.159/svn/sxt_svn">http://192.168.120.159/svn/sxt_svn</a>	http-mutil	架构课程-工程构建管理-SVN		<a href="#">设置用户组</a>	<a href="#">设置权限</a>	<a href="#">删除</a>

NO.	项目	资源	用户组/帐号	权限	删除
1	sxt_svn	[sxt_svn/]	manager	可读可写	<a href="#">删除</a>

NO.	项目	资源	用户组/帐号	权限	删除
1	sxt_svn	[sxt_svn/]	manager	可读可写	<a href="#">删除</a>

项目管理(sxt\_svn) --> 权限管理

NO.	项目	资源	用户组/帐号	权限	删除
1	sxt_svn	[sxt_svn/]	manager	可读可写	<a href="#">删除</a>

项目管理(sxt\_svn) --> 权限管理

项目管理(sxt\_svn) --> 权限管理

## 7.4.4 授权文件的变化

修改用户和权限后都会有对应的授权文件和 *MySQL* 数据信息变化。主要集中在 *MySQL* 数据库 *svnadmin.usr* 表和 *Linux* 系统 */svn/passwd.http* 文件中。

*svnadmin.usr* 表内容如下：

```
mysql> select * from usr;
+-----+-----+-----+
| usr   | psw   | role  |
+-----+-----+-----+
| *     | *     | NULL  |
| admin | cm9vdA** | admin |
| test_svn | cm9vdA** |      |
+-----+-----+-----+
3 rows in set (0.00 sec)
```

*/svn/passwd.http* 文件内容如下：

```
admin:{SHA}3Hbp8MAAbo+RngxRXGbbujmC94U=
test_svn:{SHA}3Hbp8MAAbo+RngxRXGbbujmC94U=
```