# 1安装

首先检查是否安装了低版本的 svn

 在ssh下输入命令 rpm -qa subversion

(卸载旧版本的svn 命令 yum remove subversion)

然后安装subversion

  yum install subversion (这种方式安装最简单)

再安装apache的Linux SVN模块  （这一步也需要，不知道为什么）

yum install mod\_dav\_svn

测试 SVN是否安装成功 svnserve --version  如果显示版本信息，则安装成功

# 2.配置

1. 建立版本库目录(可建立多个，新建库后以下各项都需重新配置。注意区别安装目录与版本库目录,以下讲的都是版本库目录)

mkdir –p /svn/repos

2. 建立svn版本库(与上面目录对应)

svnadmin create /svn/repos      （repos 就是版本库名）

然后我们能看的 /svn/repos 目录下生成了一堆文件

首先去到 /svn/repos/conf下面 修改文件 svnserve.conf

将里面的代码设置成如下

[general]

anon-access=none    # 使非授权用户无法访问

auth-access=write    # 使授权用户有写权限

password-db=passwd    # 指明密码文件路径

authz-db=authz    # 访问控制文件

realm=/svn/repos # 认证命名空间，subversion会在认证提示里显示，并且作为凭证缓存的关键字

配置用户信息

打开/svn/repos/conf/passwd 文件

添加用户

[users]

fxh=fxh123

fei=fei123  （此时添加了2个用户，一个是fxh，密码是fxh123; 一个是fei,密码是fei123）

再打开/svn/repos/conf/authz文件 配置用户的权限

在文件最后添加代码如下：

[/]

fxh=rw

fei=r      （此处说明fxh和fei用户对svn服务器的根目录下面的代码有操作权限 fxh有读和写的权限，fei只有读的权限）

# 开启svn服务

**svnserve -d -r /svn**

然后 到我们要进行代码修改的地方去checkout文件 比如我是到网站的根目录里 /data/www/default 创建我自己的目录mysvn

现在 /data/www/default下面有mysvn目录

在mysvn目录里面 运行命令  svn checkout svn://localhost/repos

这是在本机上checkout出 svn文件

之前一直出现svn认证出错的问题  conf下面的配置文件修改的地方行首不能有空格

# svn co svn://192.168.60.10/repos  
Authentication realm: <svn://192.168.60.10:3690> 92731041-2dae-4c23-97fd-9e1ed7f0d18d  
Password for ‘root’:  
Authentication realm: <svn://192.168.60.10:3690> 92731041-2dae-4c23-97fd-9e1ed7f0d18d  
Username: fxh  
Password for ‘pm’:  
  
Checked out revision 1.  ( 测试提取成功 )

如果svn目录不在根目录的 比如 /mnt/wwwroot/fxh/svn/comoa

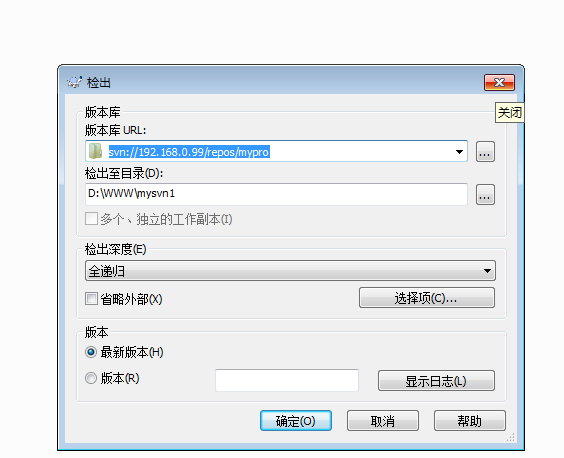
此时开启svn服务器的时候要这样运行命令 svnserve -d -r /mnt/wwwroot/fxh/svn/

如下检出

svn co svn://120.26.64.149/comoa

如果是在局域网的另一台机器（B）上要checkout以上机器的svn文件 比如以上svn服务器ip地址 是192.168.0.99

首先B机器安装客户端的svn，安装成功后 在工作目录里 鼠标右键  检出文件



其他机器检出此svn项目文件同上

我的svn仓库在/svn/repos

局域网一直连接不上

后来我在命令行执行下面的代码

svnserve --listen-port 3690 -d -r /svn/repos

给仓库项目设置端口

关闭防火墙 ，然后我本机就连接上了 ^ \_^

开启svn命令

svnserve -d -r /home/svndata

后面那个要跟你自己的数据仓库目录。

这样我们就启动了SVN了，SVN默认监听的是3690。有时候我们需要更改一下监听的端口，这时就需要设定一下监听端口了，按照如下命令来进行：

svnserve --listen-port 9999 -d -r /opt/svndata

这样我们就把监听端口改成了9999了，当然你也可以可以把9999改成你自己想监听的端口~

# svn 操作命令

1、将文件checkout到本地目录   
   
  svn checkout path（path是服务器上的目录）   
   例如：svn checkout svn://192.168.1.1/pro/domain

   简写：svn co

2、往版本库中添加新的文件   
   
  svn add file   
  例如：svn add test.php(添加test.php)

  svn add \*.php(添加当前目录下所有的php文件)

3、将改动的文件提交到版本库   
   
  svn commit -m "LogMessage" [-N] [--no-unlock] PATH(如果选择了保持锁，就使用--no-unlock开关)   
   例如：svn commit -m "add test file for my test" test.php

   简写：svn ci

4、加锁/解锁   
   
  svn lock -m "LockMessage" [--force] PATH   
  例如：svn lock -m "lock test file" test.php

  svn unlock PATH

5、更新到某个版本   
   
  svn update -r m path   
   例如：   
   svn update如果后面没有目录，默认将当前目录以及子目录下的所有文件都更新到最新版本。   
   svn update -r 200 test.php(将版本库中的文件test.php还原到版本200)   
   svn update test.php(更新，于版本库同步。如果在提交的时候提示过期的话，是因为冲突，需要先update，修改文件，然后清除svn resolved，最后再提交commit)

   简写：svn up

6、查看文件或者目录状态   
   
  1）svn status path（目录下的文件和子目录的状态，正常状态不显示）   
   【?：不在svn的控制中；M：内容被修改；C：发生冲突；A：预定加入到版本库；K：被锁定】   
  2）svn status -v path(显示文件和子目录状态)   
   第一列保持相同，第二列显示工作版本号，第三和第四列显示最后一次修改的版本号和修改人。   
   注：svn status、svn diff和 svn revert这三条命令在没有网络的情况下也可以执行的，原因是svn在本地的.svn中保留了本地版本的原始拷贝。  
   简写：svn st   
   
7、删除文件   
   
  svn delete path -m "delete test fle"   
  例如：svn delete svn://192.168.1.1/pro/domain/test.php -m "delete test file"   
  或者直接svn delete test.php 然后再svn ci -m 'delete test file‘，推荐使用这种   
  简写：svn (del, remove, rm)   
   
8、查看日志   
   
  svn log path

  例如：svn log test.php 显示这个文件的所有修改记录，及其版本号的变化

9、查看文件详细信息   
   
  svn info path   
  例如：svn info test.php   
10、比较差异   
   
  svn diff path(将修改的文件与基础版本比较)   
  例如：svn diff test.php   
       svn diff -r m:n path(对版本m和版本n比较差异)   
  例如：svn diff -r 200:201 test.php

  简写：svn di

11、将两个版本之间的差异合并到当前文件   
   
  svn merge -r m:n path   
  例如：svn merge -r 200:205 test.php（将版本200与205之间的差异合并到当前文件，但是一般都会产生冲突，需要处理一下）   
      
12、SVN 帮助   
   
  svn help

  svn help ci

------------------------------------------------------------------------------

以上是常用命令，下面写几个不经常用的   
   
------------------------------------------------------------------------------   
   
13、版本库下的文件和目录列表   
   
  svn list path   
  显示path目录下的所有属于版本库的文件和目录

  简写：svn ls

14、创建纳入版本控制下的新目录   
   
  svn mkdir: 创建纳入版本控制下的新目录。   
  用法: 1、mkdir PATH...   
       2、mkdir URL...   
   
15、恢复本地修改   
   
   svn revert: 恢复原始未改变的工作副本文件 (恢复大部份的本地修改)。revert:   
   用法: revert PATH...   
   注意: 本子命令不会存取网络，并且会解除冲突的状况。但是它不会恢复   
        被删除的目录   
    
16、代码库URL变更   
   
    svn switch (sw): 更新工作副本至不同的URL。   
    用法: 1、switch URL [PATH]   
          2、switch --relocate FROM TO [PATH...]   
   
          1、更新你的工作副本，映射到一个新的URL，其行为跟“svn update”很像，也会将   
             服务器上文件与本地文件合并。这是将工作副本对应到同一仓库中某个分支或者标记的   
             方法。   
          2、改写工作副本的URL元数据，以反映单纯的URL上的改变。当仓库的根URL变动   
             (比如方案名或是主机名称变动)，但是工作副本仍旧对映到同一仓库的同一目录时使用   
             这个命令更新工作副本与仓库的对应关系。   
    
17、解决冲突   
   
   svn resolved: 移除工作副本的目录或文件的“冲突”状态。   
   用法: resolved PATH...   
   注意: 本子命令不会依语法来解决冲突或是移除冲突标记；它只是移除冲突的   
   相关文件，然后让 PATH 可以再次提交。   
    
18、输出指定文件或URL的内容。   
   
  svn cat 目标[@版本]...如果指定了版本，将从指定的版本开始查找。   
  svn cat -r PREV filename > filename (PREV 是上一版本,也可以写具体版本号,这样输出结果是可以提交的)

svn 设置忽略文件

svn propedit svn:ignore dir

svn propset svn:ignore '\*' dir

svn commit -m 'ignore dir'

# [[svn] 命令行设置ignore目录和文件](http://www.cnblogs.com/bluefrog/archive/2012/06/27/2566279.html)

**目录**

**对当前目录下的data目录设置ignore  点表示当前目录**

svn ps svn:ignore "data" .

**指定文件**

svn ps svn:ignore "test.dat" .

**指定文件类型**

svn ps svn:ignore "\*.pyc" .

wget --no-check-certificate  <https://api.sinas3.com/v1/SAE_lnmp/soft/lnmp1.2-full.tar.gz>

wget http://ssss