**SVN 教程**

### SVN 简介

Subversion(SVN) 是一个开源的版本控制系統, 也就是说 Subversion 管理着随时间改变的数据。 这些数据放置在一个中央资料档案库(repository) 中。 这个档案库很像一个普通的文件服务器, 不过它会记住每一次文件的变动。 这样你就可以把档案恢复到旧的版本, 或是浏览文件的变动历史。

SVN 的一些概念

repository（源代码库）:源代码统一存放的地方

Checkout（提取）:当你手上没有源代码的时候，你需要从repository checkout一份

Commit（提交）:当你已经修改了代码，你就需要Commit到repository

Update (更新):当你已经Checkout了一份源代码， Update一下你就可以和Repository上的源代码同步，你手上的代码就会有最新的变更

日常开发过程其实就是这样的（假设你已经Checkout并且已经工作了几天）：Update(获得最新的代码) -->作出自己的修改并调试成功 --> Commit(大家就可以看到你的修改了) 。

如果两个程序员同时修改了同一个文件呢, SVN可以合并这两个程序员的改动，实际上SVN管理源代码是以行为单位的，就是说两个程序员只要不是修改了同一行程序，SVN都会自动合并两种修改。如果是同一行，SVN会提示文件Confict, 冲突，需要手动确认。

### SVN 安装

在windows下安装 SVN

1、准备svn的安装文件

在浏览器中打开 <http://subversion.apache.org/packages.html#windows>

1. 把svn安装目录里的bin目录添加到path路径中，在命令行窗口中输入 svnserve --help ,查看安装正常与否。至此，windows下的SVN安装完成

### SVN 生命周期

创建版本库

Create 操作是用来创建一个新的版本库。大多数情况下这个操作只会执行一次。当你创建一个新的版本库的时候，你的版本控制系统会让你提供一些信息来标识版本库，例如创建的位置和版本库的名字。

检出

Checkout 操作是用来从版本库创建一个工作副本。工作副本是开发者私人的工作空间，可以进行内容的修改，然后提交到版本库中。

更新

顾名思义，update 操作是用来更新版本库的。这个操作将工作副本与版本库进行同步。由于版本库是由整个团队共用的，当其他人提交了他们的改动之后，你的工作副本就会过期。

执行变更

当检出之后，你就可以做很多操作来执行变更。编辑是最常用的操作。你可以编辑已存在的文件来，例如进行文件的添加/删除操作

复查变化

当你检出工作副本或者更新工作副本后，你的工作副本就跟版本库完全同步了。但是当你对工作副本进行一些修改之后，你的工作副本会比版本库要新。在 commit 操作之前复查下你的修改是一个很好的习惯。

修复错误

我们来假设你对工作副本做了许多修改，但是现在你不想要这些修改了，这时候 revert 操作将会帮助你。Revert 操作重置了对工作副本的修改。它可以重置一个或多个文件/目录。当然它也可以重置整个工作副本。在这种情况下，revert 操作将会销毁待变更列表并将工作副本恢复到原始状态。

解决冲突

合并的时候可能会发生冲突。Merge 操作会自动处理可以安全合并的东西。其它的会被当做冲突。例如，"hello.c" 文件在一个分支上被修改，在另一个分支上被删除了。这种情况就需要人为处理。Resolve 操作就是用来帮助用户找出冲突并告诉版本库如何处理这些冲突

提交更改

Commit 操作是用来将更改从工作副本到版本库。这个操作会修改版本库的内容，其它开发者可以通过更新他们的工作副本来查看这些修改。在提交之前，你必须将文件/目录添加到待变更列表中。列表中记录了将会被提交的改动。当提交的时候，我们通常会提供一个注释来说明为什么会进行这些改动。这个注释也会成为版本库历史记录的一部分。Commit 是一个原子操作，也就是说要么完全提交成功，要么失败回滚。用户不会看到成功提交一半的情况。

### centos7 搭建svn服务器

1. 安装svn服务器：
2. 检查是否已经安装：
3. svn --version

]# yum install subversion

2、配置svn服务器：

建立svn版本库根目录及相关目录即svndata及密码权限命令svnpasswd；

]# mkdir -p /application/{svndata,svnpasswd}

svndata目录：作为存放svn的根目录；

svnpasswd目录：作为存放登录svn的用户和密码；

3、启动svn服务：

]# svnserve -d -r /application/svndata/

参数：

-d：表示后台运行守护模式；

-r：表示svn服务的根目录；

检测svn端口3690是否已经监听：

netstat -lntup | grep 3690

查看一下svn的进程，命令 ps -ef|grep svnserve

4、建立版本库：

在/application/svndata/目录下，创建项目sadoc：

]# svnadmin create /application/svndata/sadoc

会自动创建出一系列固定的目录：

运行创建版本库的命令，指定数据存储为 FSFS，如果要指定为 Berkeley DB，则将 fsfs 替换为 bdb

$ svnadmin create --fs-type fsfs /etc/svn/repos

]# tree /application/svndata/sadoc

主要目录说明：

conf目录：主要是配置文件、登录用户、密码认证；

hooks目录：主要是svn的钩子脚本模板，使用时去掉.tmpl后缀即可，钩子脚本实际上就是代码提交前后，可以分别执行的脚本，脚本可用perl、shell、python等语言编写，该目录里面脚本文件名代表提交代码执行的不同时机，如提交代码前执行的脚本pre-commit，提交代码后执行的脚本post-commit，更多详情自行学习；

5、修改配置文件：

]# vim /application/svndata/sadoc/conf/svnserve.conf

anon-access = none

#不允许匿名访问，read为允许匿名访问；

auth-access = write

#允许可写；

password-db = /application/svnpasswd/passwd

#密码数据库

authz-db = /application/svnpasswd/authz

#控制权限数据库

注意：配置文件必须顶格写；

把用户和密码文件放到自定义的目录中：

]# cp authz passwd /application/svnpasswd/

改权限：

]# chmod 700 /application/svnpasswd/\*

配置passwd文件：

[users]

oldboy = oldboysecret

test = testsecret

test1 = test1secret

passwd配置文件说明：

格式为：svn账号=svn密码

账号和密码可随意指定；

配置authz：

[groups]

#定义组信息，注意：用户账号在passwd里；

ett\_sa = oldboy,test1

ett\_wangguan = test

[sadoc:/]

@ett\_sa = rw

@ett\_wangguan = r

test = r

authz配置文件说明：

1.权限配置文件中出现的用户必须已经在用户配置文件中定义；

2.对权限配置文件的修改立即生效，不必重启svn；

3.一个用户组可包含多个用户，用户间用逗号分隔；

4.版本库目录格式：

[<版本库>://项目目录]

@<用户组名> = <权限>

<用户名> = <权限>

其中，方括号内部分可有多种写法：

[/]：表示根目录及以下，根目录是svnserve启动时指定的，[/]表示对全部版本库设置权限；

[repos:/]：表示对版本库repos设置权限；

[repos:/sadoc]：表示对版本库repos中的sadoc项目设置权限；

[repos:/sadoc/oldboy]：表示对版本库repos中的sadoc项目的oldboy目录设置权限；

权限主体可以是用户、用户组或\*，用户组在前面加@，\*表示全部用户；

权限可以是w、r、wr和空，空表示没有任何权限；

authz中每个参数都要顶格写，开头不能有空格；

对于组，要以@开头，而用户不需要；

6、重启svn服务：

]# pkill svnserve

]# svnserve -d -r /application/svndata/

注意：

更改svnserve.conf需要重启svn，更改authz、passwd文件时不需要重启；

检查3690端口是否处于监听状态，至此svn服务器搭建完成；

4、服务命令：

启动svnserve服务

systemctl start svnserve.service

设置开机自启动

systemctl enable svnserve.service

停止开机自启动

systemctl disable svnserve.service

查看服务当前状态

systemctl status svnserve.service

重新启动服务

systemctl restart svnserve.service

查看所有已启动的服务

systemctl list-units --type=service

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

本文来自 这个码农有点笨 的CSDN 博客 ，全文地址请点击：https://blog.csdn.net/NDF923/article/details/74171867?utm\_source=copy