

## آشنایی با Subversion بخش سوم

### نسخه کاری

این اصطلاح را قبلا هم در این بخش بارها دیده‌اید. یک نسخه کاری در واقع چیزی نیست جز یک شاخه در کامپیوتر شخصی شما که مقداری فایل را در خود نگاه می‌دارد! این فایلها در واقع کپی از فایلها موجود در مخزن فایل سرور هستند. هر کاری که دوست داشته باشید می‌توانید با فایلهای نسخه کاری خود بکنید. می‌توانید آنها را تغییر دهید، پاک کنید، اسمشان را عوض کنید، آنها را در زیر شاخه‌های مختلفی که درون نسخه کاری قرار دارد کپی کنید و خلاصه هر کاری که به ذهنتان می‌رسد با آنها انجام دهید، بالاخره کامپیوتر مال شماست و صاحب اختیار هستید. حتی اگر این فایلها مربوط به زبان برنامه نویسی باشند که کامپایل پذیر باشد، می‌توانید خیلی ساده همانطور که قبلا آنها را کامپایل می‌کردید، اینجا هم آنها را کامپایل کنید. Subversion به هیچ عنوان بدون اجازه شما این فایلها را در اختیار دیگران قرار نمی‌دهد و فایلهای دیگران را نیز در نسخه کاری شما قرار نمی‌دهد. پس از اینکه کارتان روی فایلها تمام شد، می‌توانید با اجرای یک دستور آنها را منتشر (publish) کرده و در اختیار بقیه کاربران قرار دهید. به هنگام منتشر کردن فایلهای خود نیز اگر نسخه اولیه‌ای که شما دانلود کرده بودید و این تغییرات را روی آن اعمال کرده‌اید با نسخه موجود در مخزن فایل برابر نباشد، Subversion پیغام لازم مبنی بر تصادم (Conflict) را به شما می‌دهد که با توجه به توضیحات بالا دیگر بقیه داستان را می‌توانید خود حدس بزنید.

نسخه کاری شما علاوه بر فایلهای شما، یک سری فایلهای سیستمی را نیز که برای کار Subversion لازم می‌باشد در خود نگاه می‌دارد. این فایلهای سیستمی درون شاخه‌ای به نام svn قرار دارند. در بخش بعد توضیح خواهیم داد که این فایلها چه هستند و چه فایده‌ای دارند.

### Subversion در عمل

تئوری کافیست، در این بخش که در واقع بخش آخر معرفی Subversion در این مقاله کوچک می‌باشد نحوه نصب Subversion، راه انداختن یک سرور کوچک و ساخت مخزن فایل بر روی سرور را به شما نشان می‌دهم. پس از آنکه از طرف سرور خیالتان جمع شد، نحوه کار با این سرور و ساخت نسخه کاری بر روی دستگاه کاربران و دستورات مفید مدیریت فایلها را به شما نشان داده و در آخر هم منابع تکمیلی، یا شاید بتوان گفت منبع اصلی کار با Subversion را به شما معرفی می‌کنم.

### نصب Subversion

Subversion بر روی لایه قابل حمل APR ساخته شده و این خود باعث می‌شود که Subversion بر روی هر سیستمی که Apache بر روی آن قابل نصب باشد، کار کند. بنابراین می‌توانید Subversion را روی Windows، Linux، تمام نسخ BSD و Mac OS X و Netware نصب کنید.

راحتترین روش نصب Subversion مراجعه به سایت [1] و دانلود نسخه باینری مربوط به سیستم عامل خودتان می‌باشد. این سایت معمولا بسته‌های نصاب گرافیکی برای کاربران ویندوز مایکروسافت، و پکیج‌های DEB و RPM و ... را ارائه می‌دهد. البته روشهای دیگری نیز برای نصب وجود دارد. می‌توانید بسته سورس برنامه را دانلود کرده و خود کامپایل کنید، اگر کاربر دبیان هستید می‌توانید از دستور معروف apt-get install subversion استفاده کنید و ...

### راه اندازی سرور subversion

من اکنون اینطور فرض می‌کنم که شما Subversion را بر روی دستگاه خود نصب کرده‌اید و می‌توانید دستورات svn، svnlook و svnadmin را اجرا نمایید. لازم به ذکر است که اگر هدف شما از خواندن این متن تنها آشنایی با مفهوم Subversion و کنترل ورژن باشد تا بحال اینکار را انجام داده‌اید و نیازی نیست که بیشتر از این مطالعه کنید! ولی اگر واقعا خواستار کار با این سیستم می‌باشید، حتما ابتدا این نرم‌افزار را نصب کنید و بعد اقدام به خواندن بقیه مطالب بکنید، چون در غیر اینصورت طولی نمی‌کشد که کاملا از کنترل ورژن متنفر خواهید شد، دلیل این امر هم همانند بقیه کارهای برنامه‌نویسی ساده است: احساس خواب آلودگی مفرط!

بعد از یکبار نصب Subversion و کار کردن با آن و عادت کردن به این محیط، مطمئن باشید که این سوال را بارها از خود خواهید پرسید که من تا بحال چگونه بدون آن کار می‌کردم؟! اکنون می‌خواهیم همان مخزن فایل معروف را نصب کنیم که کاربران قرار است فایلهای خود را در آن بریزند و تمام log ها و نسخه‌های مختلف در آن ذخیره می‌شوند. برای ساخت این مخزن فایل در سرور خود ابتدا باید شاخه‌ای را برای اینکار انتخاب کنید. پس از اینکه این شاخه را انتخاب کردید، بوسیله دستور زیر می‌توانید مخزن فایل خود را ایجاد نمایید:

```
$ svnadmin create /path/to/repository
```

لازم به ذکر است که قبل از اجرای دستور بالا، شاخه مورد نظر خالی می‌باشد. پس از اجرای دستور می‌توانید درون آن شاخه را مشاهده کنید:

```
$ ls /path/to/repository
conf/ dav/ db/ format hooks/ locks/ README.txt
```

همانطور که مشاهده می‌کنید چندین شاخه درون مخزن فایل ایجاد شده‌اند. بحث و بررسی تک تک این شاخه‌ها در حوصله این مقاله نمی‌باشد، تنها به ذکر این نکات اکتفا می‌کنم که شاخه config حاوی فایلهای config، شاخه db شامل فایلهای بانک اطلاعاتی Berkeley DB و شاخه hooks حاوی اسکریپتهایی می‌باشد که می‌توان قبل و یا بعد از ارسال

داده ها توسط کاربر آنها را اجرا کرد، در واقع این اسکریپتها در حکم یک سری trigger می باشند. خوب، اکنون شما صاحب یک مخزن فایل رسمی ولی خالی شده اید! معمولا ما ابتدا یک سری فایل در قالب یک پروژه داریم که بعد از بزرگ شدن پروژه نیاز به ورژن کنترل پیدا می کنند و ما به همین خاطر سراغ subversion و ساخت مخزن فایل می رویم. خوب، اگر این سناریو در مورد شما هم صادق است (قطعا همینطور است!) حال باید دید چگونه می توانید مدیریت فایل های پروژه خود را به subversion بسپارید، در واقع آنها را در مخزن فایل قرار دهید تا دیگر مخزن شما خالی نباشد! راحت ترین روش انجام این کار همان import کردن فایلها به درون مخزن فایل می باشد. فرض کنید پروژه ای دارید به نام Calender که درون شاخه ای به همین نام قرار دارد و می خواهید این پروژه را درون مخزن فایل جدیدتان قرار دهید، نحوه کار به اینصورت است:

---

```
$ svn import /home/hoomand/Calender file:///path/to/repository -m "Initial import"
```

---

همانطور که می بینید پس از دستور svn import ابتدا آدرس شاخه ای که فایل های Calender ما در آن قرار دارد را می آوریم، سپس آدرس مخزن فایل را. لطفا دقت کنید که هر وقت، تکرار می کنیم هر وقت خواستید آدرس مخزن فایل را در هر دستوری ذکر کنید، حتما آنرا با پروتکل و بصورت URL بنویسید. فراموش نکنید که مخزن فایل در واقع درون سرور قرار دارد، در اینجا که من از پروتکل <file:///> استفاده کردم به این دلیل بود که دستور import را از درون خود سرور اجرا می کردم. اگر این محیط یک محیط واقعی بود و ما در حال نوشتن یک مقاله آموزشی نبودیم، به احتمال قوی من از روی دستگاه کلاینت خود همچنین کاری را انجام میدادم، در اینصورت اگر بر فرض اسم سرور من [svn.hoomand.com](http://svn.hoomand.com) بود، آنگاه دستور بالا بصورت زیر نوشته می شد:

---

```
$ svn import /home/hoomand/Calender http://svn.hoomand.com -m "Initial import"
```

---

خوب، پس از اجرای این دستور، خروجی مشابه زیر مشاهده خواهید کرد:

```
Adding /home/hoomand/Calender/firstFile.php
Adding /home/hoomand/Calender/someOtherFile.php
Adding /home/hoomand/Calender/someGraphic.png
Adding /home/hoomand/Calender/README
Adding /home/hoomand/Calender/template/gui.tpl
Adding /home/hoomand/Calender/template/body.html
...
Committed revision 1.
$
```

در خط آخر نیز جمله معروف `Committed revision #number` آمده است که با داشتن شماره ۱ نشان می دهد که اولین نسخه خود را با موفقیت درون مخزن فایل قرار داده اید. از این به بعد کاربران می توانند به مخزن فایل متصل شده و با دانلود کردن فایل های آن نسخه کاربری شخصی خود را ایجاد نمایند و پس از هر تغییر آنرا مجدداً در مخزن فایل قرار دهند. شماره ۱ که الان شاهد آن بودید با هر تغییر که توسط کاربران روی مخزن فایل انجام شود (فایل جدیدی اضافه شود، فایلی تغییر پیدا کند و ...) یکی یکی اضافه می شود، در مورد این شماره و نحوه اضافه شدن آن بعداً به تفصیل سخن خواهیم گفت.

خوب، اکنون که دیگر واقعا یک سری فایل در مخزن فایل خود قرار داده ایم، فکر می کنید اگر دوباره وارد سرور شویم و درون شاخه ای که مخزن فایل را در آن نصب کرده ایم نگاه کنیم، چه خواهیم دید؟! بیائید دوباره دستور زیر را پس از وارد شدن به سرور اجرا کنیم:

---

```
$ ls /path/to/repository
conf/ dav/ db/ format hooks/ locks/ README.txt
```

---

چی شد؟! باز هم که همان محتوای قبلی، پس فایل های ما کجا رفتند؟ خوب، الان وقت این است که به شما بگویم فایل های شما در واقع درون فایل سیستم خود سیستم عاملتان ذخیره نمی شود، اگر اینطور بود که دیگر شما نمی توانستید مثلاً ۵ نسخه گذشته از یکی از فایل های شما را مجدداً ببینید و در تونل زمان به عقب یا جلو بروید. Subversion از یک بانک اطلاعاتی بسیار قوی به نام Berkeley DB استفاده می کند که این بانک اطلاعاتی تمامی ویژگی های یک بانک اطلاعاتی حرفه ای همچون پشتیبانی از transaction ها و index گذاری و ... را دارد. فایل های شما درون این بانک اطلاعاتی ذخیره می شود، به بیان دیگر هیچگاه قادر به دیدن فایل های خود با دستورات سیستم عاملتان نیستید. ولی اگر واقعا دلتان برای فایل های تنگ شده است و می خواهید که آنها را قبل از تهیه نسخه کاری خود ببینید، اصلاً نگران نباشید، دستور `svn list` مختص شما طراحی شده است:

```
$ svn list file:///path/to/repository
```

```
firstFile.php
someOtherFile.php
someGraphic.png
README
template/gui.tpl
template/body.html
```

می‌بینید که فایل‌های شما در مخزن ذخیره شده‌اند. یک مطلب که فراموش کردم توضیح دهم همانا استفاده از سوئیچ `m` بود به هنگام `import` کردن فایل‌ها. هر موقع که تغییرات شما به گونه‌ای باشد که بخواهد چیزی را در مخزن فایل تغییر دهد (در واقع چیز جدیدی به آن اضافه کند، چون هیچگاه هیچی پاک نخواهد شد، تنها ورژن جدیدتر اضافه می‌شود)، شما می‌توانید یک پیغام را برای ذخیره سازی مشخص کنید. این پیغام `log` شده و در آینده می‌توان دوباره آنرا مطالعه کرد تا دلیل تغییرات را متوجه شد. اگر از این سوئیچ استفاده نکنید، ویرایشگر پیش فرض سیستم عامل شما قبل از اجرای عملیات باز می‌شود تا یادداشتتان را آنجا وارد کنید. بنابر این اگر متن طولانی برای نوشتن دارید، لزومی ندارد که از این سوئیچ استفاده کنید. لازم به ذکر است که اجباری برای وارد کردن پیغام وجود ندارد، پس از باز شدن ویرایشگر پیش فرض سیستم می‌توانید بدون تایپ هیچ توضیحی از آن خارج شوید، در اینصورت `subversion` به شما ۲ امکان می‌دهد: وارد کردن مجدد توضیحات، قطع عملیات و یا اجرای عملیات بدون ثبت هیچ یادداشتی.

## بیژن هومند

[1] <http://subversion.tigris.org>