

## Subversion – verziókövetés nem csak kódnak

Íme, egy szentségtörő megoldás, kód helyett a személyes információk különféle verzióinak kezelésére.

**E**lőfordult már, hogy arra voltunk kíváncsiak, módosítottunk-e egy bizonyos fájlt egy héttel ezelőtt, és töröltük-e belőle azt az információt, amelyre épp szükségünk lenne? Előfordult már, hogy órákig böngésztük ugyanannak a fájlnak az inkonzisztens módon elnevezett másolatait a helyes verzió megtalálásának reményében? Ha egy cipőben járunk, akkor a válasz mindkét kérdésre visszhangos igen lesz. A szoftverfejlesztők ezt a kódok esetében minden bizonnyal megoldják valamely verziókövető rendszer, például CVS vagy *Subversion* használatával. De mi legyen a többi állománytípusal? A nagy meggyes rétesének receptje nyilván nem változik olyan gyakran, mint az *rpc\_init.c*, de mi legyen, ha az eredeti elvesztése nélkül egy diétás változatot is tárolni szeretnénk? Rögvest kiderül, hogy a verziókövetés többé nem csak a forrásfájlok kiváltsága. A *Subversion* sok tulajdonsága ideálissá teszi a legkülönbébb állományok verziójának kezelésére. A *Subversionnel* nyilvántarthatjuk, hogy mikor és milyen változtatásokat végeztünk egy fájlban, így azt is könnyen kideríthetjük, a fájl pontosan mit tartalmazott egy adott pillanatban. Mindezt ráadásul helytakarékos módon, hiszen a *Subversion* csak az eredeti verziótól való eltéréseket tárolja. Így amikor megváltoztatjuk a fájlt, csak a változtatások eltárolásához kell helyet biztosítanunk, nem pedig egy teljes másodpéldánynak. A CVS-től eltérően, ez a technika a bináris állományokra is működik. A *Subversion* segítségével, a fájlok több számítógépről is könnyen elérhetők. Ahelyett, hogy amiatt aggódnánk, hogy

a pénzügyi beszámoló laptopon lévő másolata tartalmazza-e az otthoni munkaállomáson tegnap éjjel végzett változtatásokat, frissítsük a fájlt *Subversionnel*, amely automatikusan a legutóbbi változatot veszi elő a verziótárból. Mivel minden verzió ugyanabban az egyetlen verziótárban található, csak ennek a helynek a biztonsági mentéséről kell gondoskodni ahhoz, hogy az adataink biztonságban legyenek.

### Minek a verzióját?

Az eddigiek tehát felkeltették az Olvasó érdeklődését, és szeretné kipróbálni, hogy a verziókövetés előnyeinek ecsetelése nem csalás, nem ámitás. Először is azt kell eldöntenünk, milyen típusú fájlt szeretnénk verziókövetésnek alávetni. Nyilvánvaló választás lehet a merevlemez teljes tartalma, de ezt nem nevezhetnénk túl gyakorlatiasnak. Ha a verziótár egy részét lokálisan tároljuk (munkapéldány), a *Subversion* teljes másolatot készít minden egyes fájlról, hogy a helyi összehasonlításnál kiderülhessen, milyen változások történtek a verziótárban lévő példányhoz képest. Emiatt a teljes merevlemez verziókövetéséhez pontosan kétszer annyi merevlemez szükséges, mint amennyi épp van. Annak sincs sok értelme, hogy a fájlrendszer nagyrészt állandó részein, például a */usr* és */opt* könyvtárakban teljes verziókezelés folyjon. Ugyanakkor azok a könyvtárak, melyekben sok az egyedi fájl és változtatás, például a */etc* és a */home* könyvtárak, ideálisak a verziókövetéshez, hiszen a változások nyomán követésének előnye valószínűleg bőségesen kárpótol bennünket az ehhez szükséges, járulékos tárhelyért. A *Subversionnel* továbbá úgy is dönthe-

tünk, hogy a verziótár teljes fastruktúrájának csak egy részéből készítünk munkamásolatot. Így nem kell munkapéldányt tartanunk a ritkán módosított fájlokból, ami általában a merevlemezigény csökkenését eredményezi, még akkor is, ha a lokálisan tárolt fájlok kétszer annyi helyet foglalnak.

### Kalandra fel!

Most pedig leheljük életre a *Subversion* a számítógépünkön. A telepítés általában rém egyszerű. Természetesen mindig ott a lehetőség a forráskód letöltésére és fordítására, de a legegyszerűbb a választott Linux disztribúcióhoz tartozó, előre lefordított bináris csomag telepítése. Szerencsére a *Subversion* már elég érett ahhoz, hogy a legfontosabb disztribúciókhoz ilyen csomag a rendelkezésünkre áll. Valójában egyről sem tudok, amihez ne lenne. A *Subversion* telepítése után létre kell hoznunk a verziótárat. Tegyük fel, hogy a saját könyvtárunkban van egy dokumentumok nevű alkönyvtár, melyet szeretnénk verziókövetésnek alávetni. Először egy üres verziótárat kell készítenünk az

```
svnadmin create
```

paranccsal. A home könyvtárban ezt például a következő módon tehetjük meg:

```
$ svnadmin create
➤ $HOME/.dokumentumok_verziotar
```

Ezután a már meglévő dokumentumokat importálni kell a verziótárba. Erre szolgál az *svn import* parancs, amit az importálandó könyvtár elérési

útvonalas és verziótár URL-je követ. Példánkban a `file://` típusú **URL** közvetlenül a verziótárra mutat. Amennyiben a verziótárat csak a számítógépünkön használjuk, a fájlok elérésének ez a legegyszerűbb módja (bár vannak más, jobb megoldások is, amelyekre a későbbiekben kitérek):

```
$ svn import $HOME/dokumentumok
↳ file://$HOME/.documentumok_
↳ verziotar
```

A parancsot végrehajtva a *Subversion* megnyitja a kedvenc szövegszerkesztőnket, amelyben megjegyzésekkel adhatunk a művelethez. Bármit is írunk ide, az a verziótár újonnan létrehozott revíziójához kapcsolódik, amit a verziótár eseménynaplójában is megtalálhatunk. Gépeljünk be valami egyszerűt, például „documents könyvtár importálása”. Miután elmentjük a szöveget és kilépünk a szövegszerkesztőből, a *Subversion* valami ehhez hasonló kimenetet produkál:

```
Adding      documentumok/
↳ file1.txt
Adding      documentumok/
↳ file2.txt
Adding      documentumok/
↳ file3.jpg
Committed revision 1.
```

Most már bátran törölhetjük az eredeti `$HOME/dokumentumok` könyvtárat, és ismét létrehozhatjuk mint munkakönyvtár, az `svn checkout` paranccsal:

```
$ rm -rf $HOME/dokumentumok
$ svn checkout file://$HOME/
↳ .dokumentumok_verziotar
↳ $HOME/dokumentumok
```

Eddig rendben is volnánk, de ha több gépről is szeretnénk élvezni a *Subversion* nyújtotta előnyöket, akkor egy kiszolgálót kell kialakítanunk. A számos lehetőség közül talán az *Apache* használata `mod_dav`-val a legjobb választás, amely a *Subversion* verziótárat a *WebDAV* protokollal teszi hozzáférhetővé. Az alap *Apache* telepítésben könnyedén működésbe hozhatjuk a *WebDAV*-ot. Először is gondoskodnunk kell a `mod_dav` és a `mod_dav_svn` modulok betöltéséről:

```
LoadModule  dav_module
↳ modules/mod_dav.so
LoadModule  dav_svn_module
↳ modules/mod_dav_svn.so
```

Ezután létre kell hoznunk egy `<Location>` direktívát, amely a verziótárra mutat. Ha pl. azt szeretnénk, hogy a `/srv/repositories/bill_documents`-ben lévő verziótárat a `http://example.net/bill/documents` URL-lel érjük el, akkor a következő direktívát használjuk:

```
<Location /bill/documents>
  DAV svn
  SVNPath /srv/repositories/
↳ bill_documents
  AuthType None
</Location>
```

Vagy korlátozhatjuk a felhasználókat, ha nagyobb biztonságot szeretnénk:

```
<Location /bill/documents>
  DAV svn
  SVNPath /srv/repositories/
↳ bill_documents
  AuthType Basic
  AuthName "Bill's Documents"
  AuthUserFile /srv/
↳ repositories/bill_documents/
↳ passwd
  Require valid-user
</Location>
```

### A Subversion használatának alapjai

Most, hogy sikeresen beüzemeltük a *Subversion*-t, vessünk egy pillantást a legalapvetőbb utasításokra.

A *Subversion* parancssoros ügyfél-programjának neve `svn`, és az összes használatos utasítást ennek segítségével adhatjuk ki. Az `svn help pl.` teljes listát ad a rendelkezésünkre álló utasításokról, de egy bizonyosról is kérhetünk információt az

```
svn help [utasítás]
```

segítségével.

Az egyik nélkülözhetetlen alapvető utasítás az `svn add`. Ha a munkakönyvtárban új fájlt hozunk létre, az nem kerül automatikusan a verziótárba. Így tartjuk a kezünkben azoknak az állományoknak a körét, amelyeknek a verzióját követni szeretnénk. Pusztá helyparazlás lenne például egy *emacs scratch* fájl a verziótárba tenni. Az `svn add`

használata nagyon egyszerű. Csak adjuk meg a kiszemelt fájlokat és könyvtárakat, és automatikusan bejegyzés készül arról, hogy ezeket a verziótárhoz kell adni – már amennyiben a megadott fájlok és könyvtárak egy valódi munkakönyvtárban találhatóak. A figyelmes Olvasónak az előző mondatban bizonyára szemet szűrt, hogy a fájlok és könyvtárak még nem kerültek be a verziótárba. Ehhez még szükség van ennek a kérésnek a végrehajtására is. Így előjegyezhetünk egy sor fájlt külön-külön, de a tényleges bejegyzést mégis egyszerre hajthatjuk végre, egyetlen revízióként, a már verzióval rendelkező, módosított fájlokkal egy kalap alatt. Hogy is van ez a végrehajtás? Ha a munkakönyvtárban módosítunk egy fájlt, a változtatás nem kerül automatikusan a verziótárba. A módosítások bejegyzését végre kell hajtani az `svn commit` utasítással. Ennek hatására az `svn` a különbségeket a verziótárba tölti és új revíziót hoz létre. Ha épp a munkakönyvtárban állunk, többnyire elég az `svn commit`, kapcsolók nélkül, mire a megváltozott fájlok bejegyzése végrehajtható, ide értve az alkönyvtárakban lévőket is. Ha nem szeretnénk az összes fájl bejegyzését végrehajtani, megtehetjük, hogy a parancssorban csak azokat adjuk meg, amelyeknél erre szükség van. Mihelyt egy fájl bekerült a verziótárba, szabadon módosíthatjuk a munkakönyvtárban, mivel a *Subversion* észleli a módosításokat, és a `commit` hatására új verziót készít. Mindazonáltal van egy megkötés. Nem másolhatunk, mozgathatunk és törölhetünk fájlokat a szokásos `cp`, `mv` és `rm` parancsokkal, mivel így a változások kikerülnek a *Subversion* látóköréből. Ehelyett a *Subversion* megfelelő utasításait kell használni: `svn cp`, `svn mv` és `svn rm`. Ezek tulajdonképpen ugyanazt csinálják, mint a szokványos megfelelőik, azzal a kiegészítéssel, hogy a verziótár megfelelő módosításait előjegyezi a következő végrehajtáshoz. Az `svn status` utasítással ellenőrizhetjük a munkakönyvtárban lévő fájlok állapotát. A status utasítás megmutatja mely fájlok nem esnek verziókontroll alá, melyek módosultak, és melyeket jegyezték elő verziókontrollra. Az alábbi példában két módosított fájlt látunk és egy harmadikat, amely nincs a verziótárban:

## Kapcsolódás OS X-ről

Rendben, bevallom, *Mac*-es vagyok, és mivel a *PowerBookom* felettébb gyakran csatlakozik az otthoni hálózatomra, számomra rendkívül fontos a fájlok elérése erről a gépről. A *Subversion* szerencsére ebben sem okozott nehézséget. Éppoly jól fut *OS X*-en, mint *Linuxon*, és a telepítése egy szempillantás alatt kész van, akár a nyílt forráskódú termékek tele-

pítésére szolgáló *Fink* csomagkezelő rendszerrel, akár bináris csomagból. Az *OS X* az automatikus verziókezelés választása esetén sem fog csalódást okozni. Az automatikus verziótár beillesztése nem több, mint a *Connect to Server...* kiválasztása a *Finder Go* menüjében, majd a verziótár *URL*-jének a megadása. Ezután a *Volumes*-ban találjuk, és a desktopon is megjelenik, mint beillesztett hálózati megosztás.

## Kapcsolódás Windows XP-ről

Használjuk játékra, munkára, a műszakilag kevésbé jól képzett kedvesünk kiengesztelésére vagy egyszerűen kedvtelésből, meglehetősen nagy az esélye, hogy hálózatunkban ilyen számítógép is található. Jó hír, hogy a *Subversion* platformfüggetlennek tervezték, és elég komoly a támogatása *Windows*on is. A *Subversion* letöltési helyein például megtaláljuk a parancsori ügyfélprogram binárisát. Ráadásul, egy imponáló *GUI*-ügyfélprogram is létezik, a *TortoiseSVN*, amely képes beágyazódni a *Windows Explorer*-be.

A *Windows XP WebFolder*ének segítségével az automatikus verziókövetés is megvalósítható. Javasolt a *WebFolder* régi, *Explorer*-kiterjesztésként megvalósított verziójának a használatát, mert a *WinXP*-ben bevezetett új verzió nem teljesen konform. Egy verziótárhoz úgy csatlakozhatunk, hogy létrehozunk egy új hálózati helyet (*network place*), és megadjuk az *URL*-t, majd kettősponttal elválasztva a portszámot. Az *example.com/documents* verziótárhoz például a *http://example.com:80/documents URL*-t kellene begépeelnünk.

```
$ svn status
?      .GroceryList.txt.swp
M      Frogs.png
M      GroceryList.txt
```

A munkakönyvtárat az *svn update* utasítással frissíthetjük a legújabb revízióra. Ha a *Subversion* egyetlen számítógépen használjuk, a frissítést valójában felesleges. Ha azonban a verziótárat több számítógépről is használják, az *svn update* utasítást végre kell hajtani, hogy a más számítógépeken végzett és bejegyzett módosítások hozzánk is eljussanak.

## Automatikus verziókövetés

A *Subversion*nel történő verziókövetés rögzös útjának megjárása után térjünk rá az automatikus verziókövetésre. Ha a *Subversion* verziótárának kiszolgálója *Apache*, a fájlok kiolvasása és beírása a *WebDAV* protokollal történik. Ennek egy érdekes kísérőjelensége, hogy a legtöbb operációs rendszer képes a megosztott *WebDAV* könyvtárak hálózati fájlrendszerként való beillesztésére, hasonlóan a *Samba*-hoz vagy az

*NFS*-hez. Tehát megtehetjük, hogy a távoli verziótárat a helyi fájlrendszerbe beillesztjük, és munkakönyvtár létrehozása nélkül közvetlenül hozzáférhetünk az ott lévő állományokhoz. Ennek a megoldásnak van néhány olyan előnye, amely kifejezetten alkalmassá teszi arra, hogy a személyes fájlok verziókövetését így végezzük. Egyrészt, minden mentéskor új verzió készül. Anélkül valósul meg a módosításaink naplózása, hogy a *commit* utasítás végrehajtásával egyáltalán foglalkoznunk kellene. Büntetlenül létrehozhatunk új fájlt, másolhatunk, mozgathatunk és törölhetünk a szokásos parancsokkal. Végül, ha több számítógép is használja a verziótárat, biztosak lehetünk abban, hogy az *svn update* nélkül is mindig a legfrissebb változatot kapjuk. Persze az automatikus verziókövetésnek is megvannak a maga árnyoldalai. Először is meglehetősen gyors hálózati kapcsolatra van szükség a kiszolgáló és a mi számítógépünk között, ami nem feltétlenül szerencsés egy otthonától gyakran távol lévő laptopnál – bár, ez kikerülhető, ha a gyors hálózati kapcsolat fennállásakor másolatot készítünk a szükséges fájlokról, aztán később visszamásoljuk azok módosított változatait a kiszolgálóra. Másodszor, a fájloknak csak a legfrissebb változata elérhető. Ha régebbi verzióra van szükség, azt vagy egy könyvtár adott verziójának letöltésével:

```
$ svn checkout -r 1563
➤ http://$MY_SERVER/docs/pics/
```

vagy az *svn cat* utasítással oldhatjuk meg, amellyel egy bizonyos fájl adott verziója lekérhető:

```
$ svn cat -r 1563 http://
➤ $MY_SERVER/docs/pics/beach.jpg
```

Ha az automatikus verziókövetést a verziótár beillesztésével szeretnénk használni, telepíteni kell a *davfs*-t. Mihelyt kész, a verziótár beillesztése már gyerekszerű, mindössze a *mount.davfs* parancsot kell futtatni, a következő példa szerint, amelyben a *\$MY\_SERVER/docs* verziótárat a */mnt/documents* könyvtárba illesztjük:

```
$ mount.davfs http://$MY_
➤ SERVER/docs/ /mnt/documents
```

A verziókövetést természetesen az *Apache*-ban is be kell kapcsolni, amihez a *Subversion*höz készített *<Location>* direktívában szerepelnie kell az *SVNAutoVersioning* on kapcsolónak is.

## Összefoglalás

A *Subversion* rendkívül okos jószág. Bár a képességeinek a jelentős részével nem is foglalkoztunk, ennyi tudás birtokában már el lehet kezdeni az állományaink verziókövetését. Jómagam is ezt használom egy ideje, és nagyon hasznosnak bizonyult. Különösképp ez a véleményem az automatikus verziókövetésről, amely szinte észrevétlenül integrálódik a fájlrendszerbe.

*Linux Journal* 2006., 143. szám

A cikk forrásai:

➤ [www.linuxjournal.com/article/8751](http://www.linuxjournal.com/article/8751)

William Nagel  
bill@williamnagel.net