

Versionshantering

Eric Elfving
Institutionen för datavetenskap (IDA)

4 september 2015

Har du någon gång använt funktionen
"Spara som"?

- Ett versionshanteringssystem håller koll på ändringar på en uppsättning data (i vårt fall källkod) över tid.
- Hanterar ett arkiv av filer (repository eller repo)
- Alla deltagare i projektet arbetar lokalt på en egen arbetskopia och låter systemet sammanföra ändringar.
- System brukar delas in i två grupper: server-klientsystem och distribuerade system. Vi kommer presentera två olika verktyg, subversion och git.

- En server håller koll på alla filer och ändringar
- Klienter hämtar filer (eller ändringsinformation) från servern, man jobbar lokalt och skickar sedan in sina ändringar till servern
- Servern kontrollerar eventuella krockar
- Vad händer om servern kraschar?
- Normalt sett har alla samma rättigheter:
 - Bara de som har tillåtelse får göra ändringar
 - **Alla** som har tillåtelse får ändra på **allt**

- Skapades i början av 2000-talet för att ersätta ett tidigare system, CVS, som var ganska buggigt
- Hanteras av Apache sedan 2010
- Har varit med länge, finns massor av information!
- Förkortas (och omnämns ofta som) SVN

Hur använder man SVN?

- Eftersom SVN är ett server-klientsystem måste ett arkiv först skapas.
- Ofta sätts det upp på en server men det går även att skapa ett lokalt arkiv:

```
svnadmin create <sökväg>
```

- När man har ett arkiv kan man hämta ut en lokal arbetskopior:

```
svn checkout <adress>
```

(<adress> byts ut mot `file://<absolut sökväg>` om man har ett lokalt arkiv)

- När du ska koda i ett projekt med subversion börjar du med att uppdatera arbetskatalogen:

```
svn update
```

- Alla filer som uppdaterats av någon i projektet ändras i min lokala katalog (systemet frågar om det skulle bli krockar)
- Därefter är det bara att koda som vanligt
- För att lägga till filer skapar du dem som vanligt och säger åt svn att versionshantera dem:

```
svn add <fil> [<fil>*]
```

- När du känner dig nöjd skickar du in dina ändringar:

```
svn commit -m "Beskrivande text om ändringarna"
```

- Endast ändringar skickas (åt båda hållen) - tidsåtgång beror på antal ändringar, inte kodstorlek
- Alla commits är fullt atomära - om något går fel händer inget
- Kommandon kan vanligen förkortas (up istället för update, ci=commit osv)

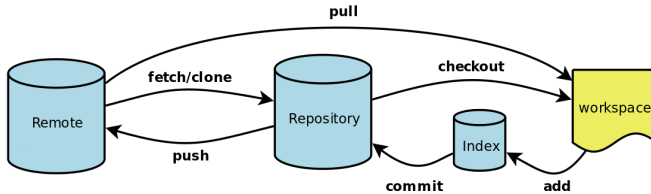
- Arbetar med en peer-peer (p2p) modell istället för en central server
- Alla arbetskopior håller koll på alla ändringar
- För att samarbeta kopierar man (klonar) hela arkivet
- Alla som hämtat ut en kopia kan göra vilka ändringar de vill
- Projektets ledare bestämmer vilka ändringar som ska sparas
- Fungerar normalt mycket bra för större projekt
- Eftersom allt är lokalt utförs commits ofta snabbare

- Git skapades av Linus Torvalds för att hantera linux-kärnan
- Namnet kan betyda lite olika. Följande säger Linus själv:

I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'git'.
- Han ger även dessa förklaringar:
 - Random three-letter combination that is pronounceable, and not actually used by any common UNIX command.
 - Stupid. Contemptible and despicable. Simple. Take your pick from the dictionary of slang.
 - "Global information tracker": you're in a good mood, and it actually works for you. Angels sing and light suddenly fills the room.
 - "Goddamn idiotic truckload of sh*t": when it breaks
- Ett designmål: Ta CVS som exempel på hur man inte ska göra.

- Antingen skapar du ett arkiv med kommandot
`git init`
- Eller hämta ett arkiv som någon annan har skapat:
`git clone <adress>`
- Arbeta som med SVN: koda + commit
- När du känner dig nöjd med dina lokala ändringar kan du skicka upp dem:
`git push ...`
- Git jobbar mycket med grenar (branch), bara vissa får pusha till huvudgrenen (master)

- För att underlätta samarbete (och säkerhet) har man ofta en server för att hålla koll på ett git-arkiv
- Man kan installera själv men det finns gratisresurser
 - Egna projekt: GitHub (github.com), ett system för fri källkod med massor av hjälpmedel såsom kodgranskning och wiki
 - Skolprojekt: gitlab (gitlab.ida.liu.se): Du kan skapa flera projekt och på ett enkelt sätt lägga till studenter som får läsa/ändra kod.



- Subversion
 - `svn help` [kommando]
 - Välanvänt - man hittar lätt träffar på google
- Git
 - `man git`
 - `git help` [kommando]
 - <http://git-scm.com/documentation>
 - <https://github.com/>
 - <https://gitlab.ida.liu.se/>

www.liu.se