A Practical Introduction to Version Control Systems

A random CAKES(less) talk on a topic I hope others find useful!

Andrew Brampton
a.brampton@lancs.ac.uk

4th February 2009

Outline

1. Version Control Systems
   - What is Version Control
   - Basic Principles
   - Versioning Models
2. Centralised - SVN
   - Overview
   - Commands
3. Decentralised - Git
   - Overview
   - Commands

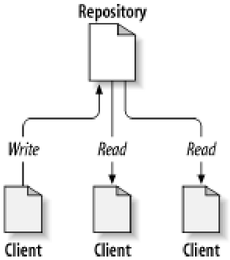
What can version control do for you?

- Backup your work
- Rollback changes
    - The original time machine
- Collaborate with others
    - Multiple users can share and edit documents
- Access documents online
    - If your repository is online, you can easily access your files anywhere

The Repository

Description

- Where all the files and directories are stored
- Stores the histories of all your files and directories
- Users read and write files to it

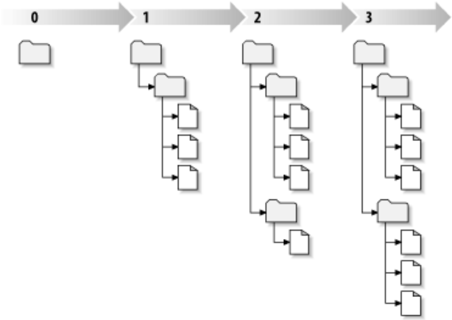


The Working Copy

Description

- A local copy of the repository
- Where your changes are made before sending them to the repository
- Conceptually the same as a sandbox

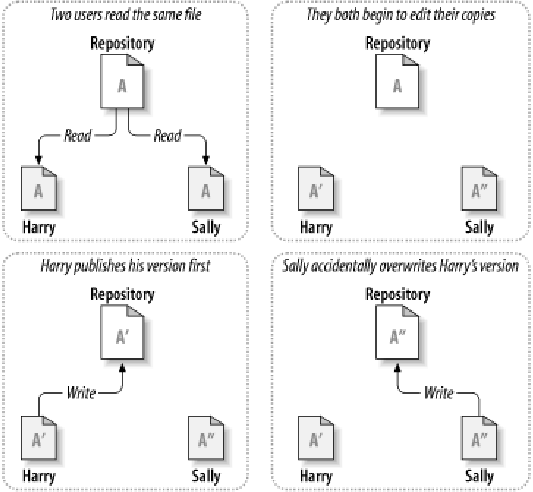
Revisions



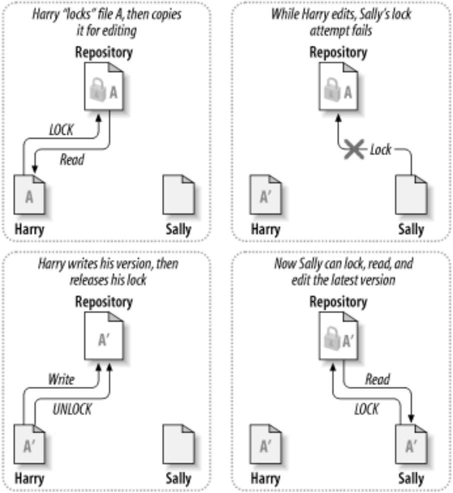
Description

- A single snap shot in time of what the repository looked like
- Global or File based revisions

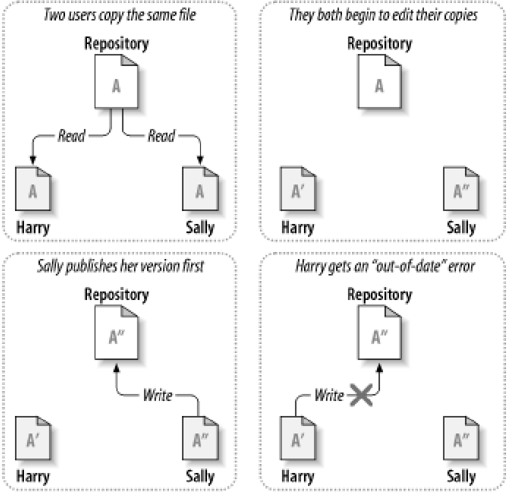
Problem of File Sharing


Two users read the same file
Repository
A
Read
Read
A
A
Harry
Sally
They both begin to edit their copies
Repository
A
A′
A″
Harry
Sally
Harry publishes his version first
Repository
A′
Write
A′
A″
Harry
Sally
Sally accidentally overwrites Harry's version
Repository
A″
Write
A′
A″
Harry
Sally

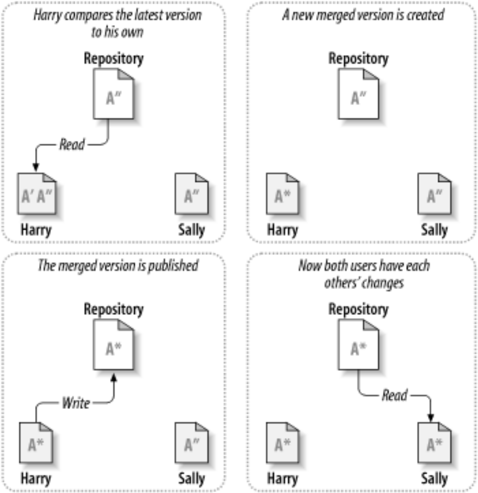
Lock-Modify-Unlock Solution


Harry "locks" file A, then copies it for editing
Repository
A
LOCK
Read
A
Harry
Sally
While Harry edits, Sally's lock attempt fails
Repository
A
Lock
A′
Harry
Sally
Harry writes his version, then releases his lock
Repository
A′
Write
UNLOCK
A′
Harry
Sally
Now Sally can lock, read, and edit the latest version
Repository
A′
Read
LOCK
A′
Harry
A′
Sally

Copy-Modify-Merge Solution


Two users copy the same file
Repository
A
Read
Read
A
A
Harry
Sally
They both begin to edit their copies
Repository
A
A′
A″
Harry
Sally
Sally publishes her version first
Repository
A″
Write
A′
A″
Harry
Sally
Harry gets an "out-of-date" error
Repository
A″
Write
A′
A″
Harry
Sally

Copy-Modify-Merge Solution


Harry compares the latest version to his own
Repository
A″
Read
A′ A″
Harry
A″
Sally
A new merged version is created
Repository
A″
A*
Harry
A″
Sally
The merged version is published
Repository
A*
Write
A*
Harry
A″
Sally
Now both users have each others' changes
Repository
A*
Read
A*
Harry
A*
Sally

Types of Version Control

- Centralised
  - Client-Server system
  - Repository stored on a server
  - *e.g* Subversion (SVN), Concurrent Versions System (CVS), Visual Source Safe, etc
- Decentralised
  - Fully decentralised, no server
  - Each user has a copy of the full repository
  - *e.g* Git, BitKeeper, Mercurial, Monotone, SVK

Subversion (SVN)

- Began in 2000 as a replacement for CVS
- Used by, Apache Software Foundation, KDE, GNOME, FreeBSD, GCC, Python, Django, Ruby, Mono, SourceForge.net, *etc.*
- Centralised / Client-Server system
  - Can be accessed via HTTP (using Webdav), A custom SVN protocol, or local filesystem
- Clients
  - Simple command line (http://subversion.tigris.org/)
  - TortoiseSVN (http://tortoisesvn.tigris.org/)
  - RapidSVN (http://rapidsvn.tigris.org/)

Create a repository

- Typically done by a admin
- You need to either setup a HTTP server, or a SVN server
- But, luckily Steven Simpson has done a lot of the work for us:
  - https://forge.comp.lancs.ac.uk/

Basic Commands

Create your working copy

```
svn checkout https://forge.comp.lancs.ac.uk/svn-repos/cakes-test/
```

Add documents to the working copy

```
svn add filename
```

Commit your changes to the repository

```
svn commit
```

Update your working copy from the repository

```
svn update
```

Advanced Commands

Show the differences you have made in your working copy

```
svn diff
```

Undo all changes in your working copy

```
svn revert
```

See a history of commits

```
svn log
```

See who is responsible for each line of a file

```
svn blame
```

Git

- Started by Linus Torvalds in early April 2005, and was usable by mid-June 2005.

  “I’m an egotistical bastard, and I name all my projects after myself. First Linux, now git”

- Used by, Linux kernel, Perl, Samba, X.org Server, Ruby on Rails, VLC, Wine, DragonFly BSD, *etc.*

- Fully Decentralised
  - Can be accessed via HTTP, FTP, rsync, or a Git protocol

- Clients
  - Simple command line (http://git-scm.com/)
  - TortoiseGit (http://code.google.com/p/tortoisegit/)

Create a repository

Everyone has a local repository

Create one from scratch

```
git init
```

Create one from an existing repository

```
git clone
git://git.kernel.org/pub/scm/linux/kernel/git/torvalds/linux-2
```

Basic Commands

Add document to your local repository

```
git add filename
```

or (for git to prompt you for each change)

```
git add -p
```

Commit your changes to your local repository

```
git commit
```

Show the differences you have made in your working copy

```
git diff
```

See a history of commits

```
git log
```

Basic Commands

Update your working copy from another repository

```
git pull
```

or

```
git pull url-to-another-repo
```

Send your changes to another repository

```
git push
```

or

```
git push url-to-another-repo
```

Thank you for listening. Any questions?

A Practical Introduction to Version Control Systems

A random CAKES(less) talk on a topic I hope others find useful!

Andrew Brampton
a.brampton@lancs.ac.uk

4th February 2009