CSC207 - Subversion

Ilir Dema

Summer 2015

Version Control Systems

Version control is used to manage the changes to code or other documents over time.

In this course we will use Subversion (svn).

There are other version control systems: CVS, Perforce, Microsoft Visual SourceSafe, etc.

And newer, better, more complex ones, such as Git and Mercurial.

The first company you end up working will likely use a version control system.

Other CSC courses (e.g., CSC209, CSC301, CSC369) use version control for assignment submission.

Why do we need version control?

Scenario I: working alone

- Need to keep track of changes

Scenario II: working in a team

- Need to coordinate between team members

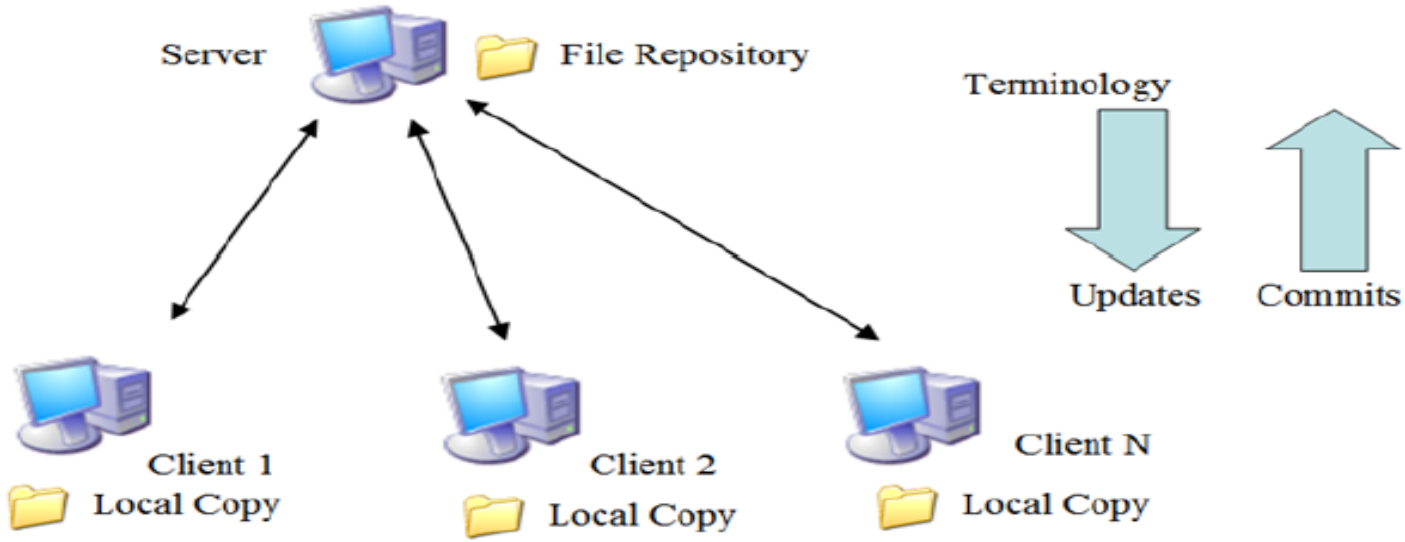
Scenario III: moving around

- Need to coordinate the mods between different locations

Flavours ofversion control: centralized and distributed.

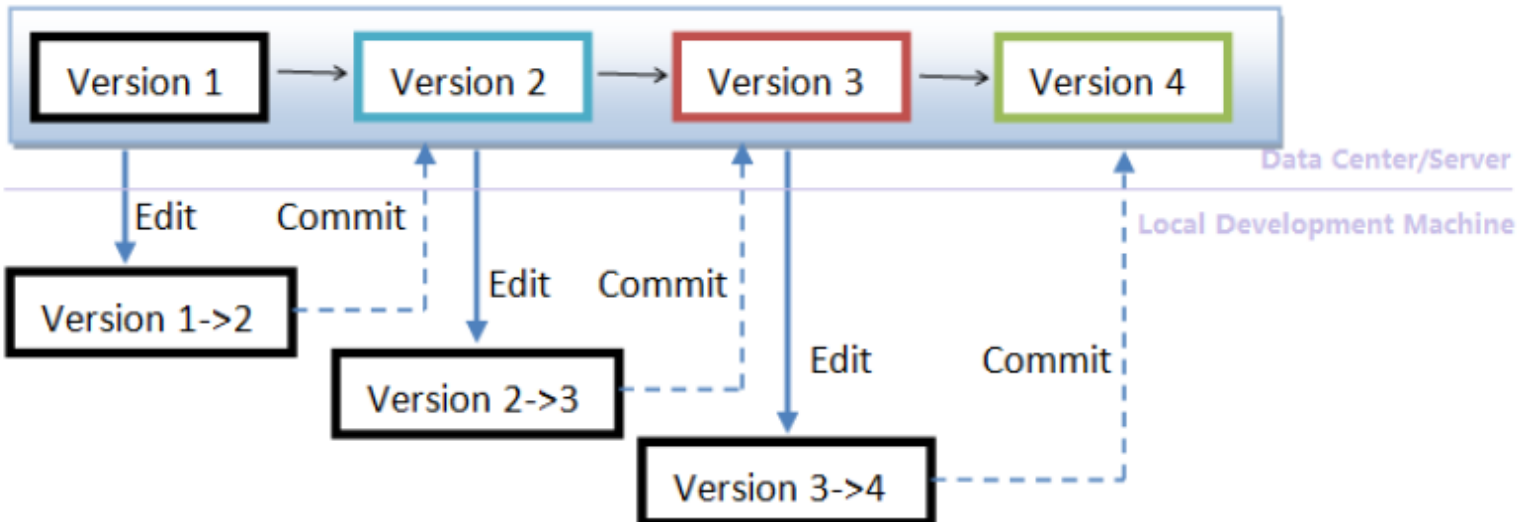
Centralized Version Control

- Keep code in a central location ("repository")
- This is the "master copy"
- Never directly modify this directory!
- Create a local copy of the repository in your account at school, on your machine at home, on your laptop...
- When the local copy changes, "commit" the changes to the repository.



Version Control - Tracking changes when working solo

- When you get something working, or you want to try something making sure you can revert to the current version, or for whatever other reason, commit the changes
- Tools allow you to revert to a previous version
- Write good log messages so that you don't have to remember what changed in each version



subversion - some common commands

- `svn checkout [url]` Get initial copy
  - Do this anywhere to get a local copy of a repository.
- `svn add [filenames]` Add new files
  - notifies Subversion that you want it to track the new files
  - add will almost always be followed by a commit because add doesn't actually modify the repository
- `svn status [filenames]` See what's changed

subversion - some common commands

- `svn update [filenames]` Synchronize with repository
  - Copies stuff from the master repository to your local copy
  - Any commits made by another person or commits done by you from another local copy will be updated in your local copy
  - Does not change the repository
  - Watch the messages closely
- `svn commit [filenames]` Commit local changes
  - Copy changes to the repository
  - Will only be allowed if the local copy is up to date
  - Creates a new version

subversion - some common commands

- `svn remove [filenames]` Remove files from repository
  - Need to commit, just like for add, to make changes to the repository
- `svn diff [filenames]` Show diffs between local copy&repo
  - Handy to see what changed between the revisions
- `svn log [filenames]` Show history of files
  - Shows log message, timestamps, and who made the revisions
- `svn tree [filenames]` Show history of files
  - Shows log message, timestamps, and who made the revisions
- `svn revert --recursive` Undo unwanted add
  - Use it before committing.

version control - storage scheme

- Storing an entire copy of a file on evey change would require an enormous amount of disk space and would make synchronization slow.
- Version control systems store incremental difference
- The incremental differences allow the system to reconstruct previous versions.

Dude, where is my repo?

You need to know the URL so you can checkout your repository:

```
svn checkout URL
```

Your repo name is the same as your CDF name. The URL will be announced from the course web site next week. You will practice svn in your first lab.