

Source code management

- **Subversion (SVN): version control system**
- **for managing large projects with multiple people**
 - widely used, open source
 - works across network as client-server
 - fixes many of shortcomings of CVS
- **store and retrieve all versions of all directories and files in a project**
 - usually source code
 - also documentation, tests, binaries, ...
- **support multiple concurrent users**
 - independent editing of files
 - merged into single version
- **highly recommended for COS 333 projects!**
 - save all previous versions of all files so you can back out of a bad change
 - log changes to files so you can see who changed what and why
 - mediate conflicting changes made by different users -
 - keeps consistency

Basic sequence

- **create a repository**
 - where SVN stores its copies of your files
 - including all changes made by anyone
- **each person checks out a copy of the files**
 - "copy - modify - merge"
 - get files from repository to work on
 - does not lock the repository
 - make changes in a local copy
 - when satisfied, check in (== commit) changes
- **if my changes don't conflict with your changes**
 - SVN updates its copies with the revised versions
 - edits on different lines are merged automatically
 - keeps previous copies
- **if my changes conflict with your changes**
 - e.g., we both changed lines in the same part of file,
SVN doesn't permit the checkin
 - we have to resolve the conflict manually

Basic sequence, continued

- **when changes are committed, SVN insists on a log message**
 - strong encouragement to record what change was made and why
 - can get a history of changes to one or more files
 - can diff versions of a file
- **can create multiple branches of a project**
- **can tag snapshots for e.g., releases**
- **can be used as client-server, so can do distributed development**
 - repository on one machine
 - users and their local copies can be anywhere

Getting started

- **to put code under SVN control, do this once:**

```
svnadmin create repository
svn import proj.dir file:///repository \
    -m 'initial repository'
svn checkout file:///repository working.dir
```
- **create, edit files in working.dir**

```
cd working.dir
ed x.c    # etc.
svn diff x.c
svn add newfile.c
```
- **update the repository**

```
svn commit # commit all the changes
```
- **for more info, read svn.help on web page, SVN book, etc.**