Source Code Control Systems

CVS (Concurrent Versions System)
PRCS (Project Revision Control System)

Manage changes to files

- Applies to entire directory tree
  - any file formats (C/Java/Latex source, images…)
- Support for tracking third party changes
  - while making local modifications
- Support for fixing bugs in old releases
- Can produce patch files

Copy-Modify-Merge

- Copy directory subtree without locking
- Make and test local modifications
- Merge changes committed by others
- Test merge
- Commit changes

CVS

- Very widely used
- Version 1.10 in /fs/unsupported
  - version 1.9 in /usr/imports
- Excellent distributed client/server system
- Based on RCS
  - makes things a bit ugly
- Hard to move, rename, delete files

CVSROOT and cvsinit

- setenv CVSROOT /fs/aufait/pugh/cvs
  - location where all repositories will be created or checked out from
  - can also be specified on command line
    - cvs -d /fs/aufait/pugh/cvs
- run cvsinit
  - initializes files in CVSROOT

Importing a project

- Prepare a clean project
  - delete files you don’t want to have tracked (.cvsignore is used, so don’t need to rm *.o)
- cd to directory at top of project
- cvs import repository vendortag releasetag
  - repository - project name
  - vendortag - Name of vendor (foobar will do)
  - releasetag - vendor version number (start will do)

Checking out a project

- cd to the directory in which you want the directory for the project to be created
- Do `cvs checkout` *repositoryName*
- Will create directory and files

Modify files to your hearts content

- Don’t *have* to worry about changes being made by others
  - – but should worry a little
- cvs diff -c shows what you have changed

commit your changes

- `cvs commit` *filenames*
  - if filenames not specified, all changed files in current directory and below
- Will be asked for a description of the change
  - make it useful and global

Merging

- If another user committed changes
  - you will get an error when you try to commit
- Use `cvs update` to get cvs to fold in those changes
  - If changes to disjoint files, or disjoint regions of files, files will be updated/patched
- If changes overlap, you get a merge conflict
  - Have to resolve manually (with text editor)

Resolving conflicts

- Your old version is stored as
  - #file.revision
- The partially merged version is:
  - <<<<<< filename
    your changes
    here
    ======
    committed changes here
    >>>>>> 1.9

.cvsignore

- Tells the system which files are never checked in
  - already includes things such as:
    - core *~ *.o
- Sources:
  - $CVSROOT/CVSROOT/cvsignore
  - ~/.cvsignore
  - .cvsignore in each directory

Adding files

- `cvs add` filenames
  - files get marked as files to be added when changes are committed
- forgetting to add files you’ve added is a frequent cause for breaking the build

CVS commands

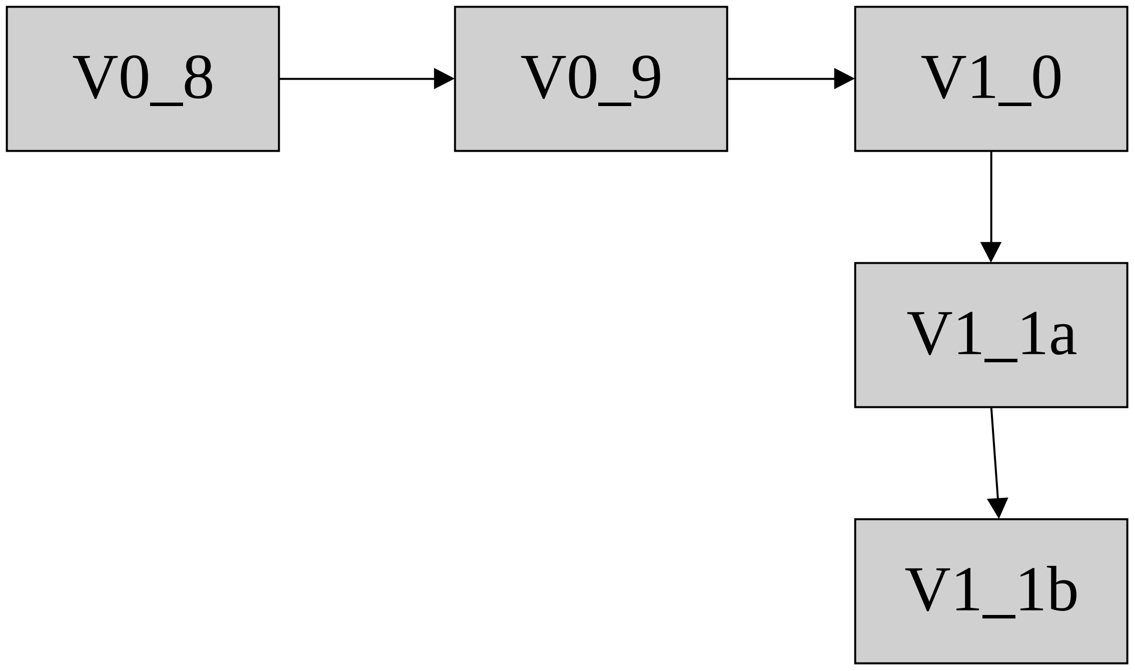
- do cvs -H command for more info
- cvs checkout
- cvs commit
- cvs diff
- cvs export
- cvs history
- cvs import
- cvs log

More cvs commands

- cvs rdiff - patch format diffs
- cvs release - indicate that repository no longer in use
- cvs rtag - add a symbolic tag to module
- cvs tag - add symbolic tag to checked out verion of files
- cvs update

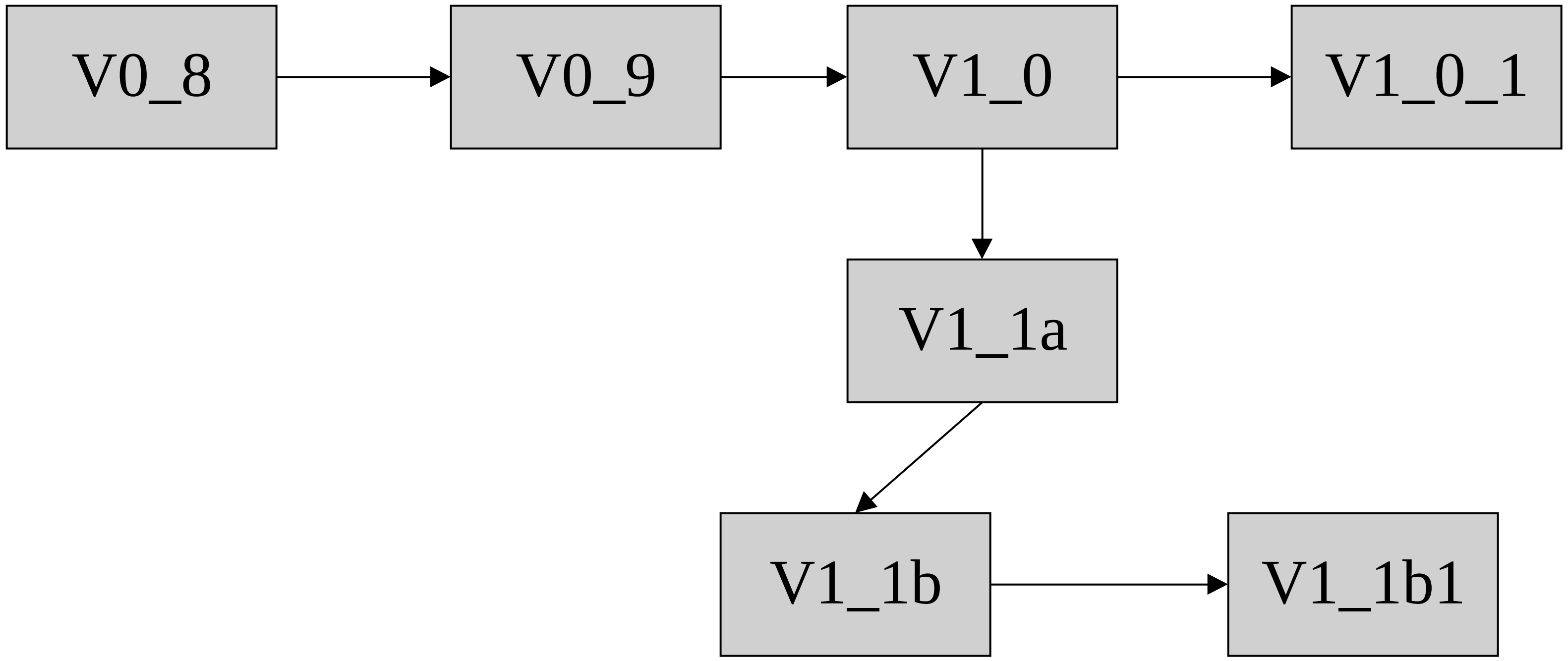
Can create branches

- Versions can form tree (Not a Dag)



Coping with branches

- Can merge changes from one branch of tree into another



Version names

- Each version of a file gets a RCS version number
  - 1.1, 1.2, 1.3
- Can use cvs commit -r 3.0 to force all file version numbers to be upgraded to version 3.0
- Tags are symbolic names that apply to a group of files

Tag names

- Can’t have . in tag names
- cvs tag release-1-0
- commands such as checkout and diff can use -r tagname

Client/Server

- Later versions of CVS allow client/server setup
- Several different methods:
  - – Use rsh

```
cvs -d :ext:wp73101@marlowe:/home/wp73101/cvs checkout cmsc731
```

  - – Use a direct connect
    - requires modifying /etc/inetd.conf

PRCS

- New source code control system by Paul Hillfinger at Berkeley
- Operations are on project version
- Not built on RCS commands
- Allows easy renaming, deletion, reintroduction of files
  - Project contains a mapping from file names to contents