

6.270 CVS Lecture

- The 6.270 Locker
- CVS

6.270 CVS Lecture - IAP 2001

1

The 6.270 Locker

- % add 6.270
- % cd /mit/6.270/Teams/*nn*
- Put “add 6.270” in ~/.environment

6.270 CVS Lecture - IAP 2001

2

CVS! (Concurrent Version System)

- Everything you really need to know is at http://www.loria.fr/~mollie/cvs/doc/cvs_toc.html
It's linked to from the 6.270 Contestant Info page.

6.270 CVS Lecture - IAP 2001

3

Why CVS?

- Multiple users want to edit the same files.
- CVS handles merging changes, even to the same file.
- CVS is no replacement for management!
- CVS keeps track of older versions
- CVS is no replacement for management!

6.270 CVS Lecture - IAP 2001

4

CVS: Basic Concept

- Repository: Where condensed versions of every version of your files are kept.
- You don't work from the repository. Rather, you work from a “checked out” copy.
- In fact, you should never manually touch anything in the repository.

6.270 CVS Lecture - IAP 2001

5

CVS: What's in a repository?

- Source files
- Makefiles
- maybe some text documentation
- NOT object files, and other binaries or executables.

6.270 CVS Lecture - IAP 2001

6

CVS: Sample Usage

- 1) Create a repository (done once)
- 2) Do a checkout (done once per user)
- 3) Add/Edit a file
- 4) "Update" your checked out files
- 5) "Commit", aka "Check In", a file
- 6) Go away for a while
- 7) Update again and return to step 3

6.270 CVS Lecture - IAP 2001

7

How CVS Works (Roughly)

- Uses "diff" to find difference between files. Thus, it only works well on line-based text files, *not* binary files.
- Uses various algorithms to figure out how to merge differences.
- Sometimes fails! You'll have to merge changes manually in those cases.

6.270 CVS Lecture - IAP 2001

8

CVS: Creating a Repository

- % setenv CVSROOT /mit/6.270/Teams/*nn*/repository
- % cvs init
- cd to your working directory. We suggest that you create a set of /mit/6.270/Teams/*nn*/*<username>* directories to work from.
- % mkdir Robot; cd Robot
(You're now in *<workingdir>/Robot*)
- % cvs import -m "Creation" Robot *teamm* start
- % cd ..
- Do a checkout

6.270 CVS Lecture - IAP 2001

9

CVS: Before you use

- CVS is in the gnu locker:
- % add gnu
- You might want to put "add gnu" in your *~/.environment* file

6.270 CVS Lecture - IAP 2001

10

CVS: Note on repository creation

- The Robot directory is called a "module".
- The "*teamm*" and "*start*" parameters to "import" are just info tags and can be arbitrary, but must be there.
- When you use the import command, any files in that directory will automatically be imported into the module after it is created. If your directory is empty, the module will simply be created.
- If you choose to import your previous files *en masse* in this way, be sure to make clean, i.e., clear your directory of all binary/class files and tool-generated .java files, before running import.

6.270 CVS Lecture - IAP 2001

11

CVS: Checking Out a Repository

- You only need to do this once per user.
- % setenv CVSROOT /mit/6.270/Teams/*nn*/repository
- cd to your working directory
- % cvs co Robot
- % cd Robot
- Work from within this directory

6.270 CVS Lecture - IAP 2001

12

CVS: Updating your files

- First cd to the directory with your files.
- % cvs update
- To get new subdirectories, use % cvs update -d
- If you get a merge conflict, you'll have to resolve it manually. Delete the version you don't want.

6.270 CVS Lecture - IAP 2001

13

CVS: Adding and Committing

- Create the file or subdirectory.
- % cvs add <filename>
- % cvs ci -m "message" <filename>
- Use commit after adding or editing a file.
- Omitting the filename commits everything (recursively).
- Not using -m prompts you for a message.

6.270 CVS Lecture - IAP 2001

14

CVS: When to Commit?

- No hard and fast rule.
- General rule of thumb:
Make sure everything compiles before you check in.
There's nothing more annoying than having your code cease to compile after checking out other someone else's changes.
- There are special exceptions.

6.270 CVS Lecture - IAP 2001

15

CVS: Removing a file

- % rm <file>
- % cvs rm <file>
- % cvs commit <file>
- Note the sort of odd ordering (which prevents you from using tab-completion)

6.270 CVS Lecture - IAP 2001

16

CVS: Removing a subdir

- Remove everything (and from CVS) inside the subdir. Now, from its parent directory, % cvs update -P
- That will get rid of any empty subdirs.

6.270 CVS Lecture - IAP 2001

17

CVS: Diffing

- Show log:
% cvs log <file>
- Differences from repository:
% cvs diff <file>
- Differences from some version:
% cvs diff -r<version> <file>
- A different display style:
% cvs diff -u <file>

6.270 CVS Lecture - IAP 2001

18

CVS in Emacs

- C-x,v, v = Commit
- C-c, C-c after entering a comment.
- See C-x,v,C-h for a list of relevant commands.

6.270 CVS Lecture - IAP 2001

19

CVS: And More

- Tagging and branching allows you to set checkpoints and such. See the French page for more information.

6.270 CVS Lecture - IAP 2001

20

How to use CVS *well!*

- CVS is no replacement for management!
- Minimize working on the same file at the same time.
- If you do, work on different portions.
- You may want to use lots of auxiliary files for different functions instead of one mondo file. (This may be more relevant in general than for 6.270 in particular.)

6.270 CVS Lecture - IAP 2001

21

CVS Pitfalls

- Never muck with the repository manually.
- Try not to make drastic changes at once.
- Always update before editing. Managing conflicts earlier is easier.
- Watch out for emacs auto-tabbing. Different versions sometimes tab differently. CVS doesn't like to merge a retabbed file.

6.270 CVS Lecture - IAP 2001

22