Version control

Introduction

- Version control tracks changes to files (what, when, who, why)
- Often used with program source code
- Also works for other text files e.g. LaTeX
- Is far more useful than it sounds ...

VC can help with:

- `rm -r *`
- I've changed my program, it's not working and I need to put it back how it was
- It was working last week, why has it broken?
- `myprog.f90 myprog.f90_old myprog.f90_v1, myprog_f90_prev`

Collaborative work

- Bug fixes and updates
- Why has that been changed?
- Who wrote that code?
- Avoid or cope with simultaneous revisions

Version control systems

- Revision Control System: RCS
- Concurrent Version System: CVS
- There are others: e.g. Subversion
- Torus and SPH code use CVS

RCS: Revision Control System

- Version control from the 1980s
- Not as many features as CVS but easy to set up
- Easy way to version control your own files
- `man rcsintro`

RCS overview

- Stores files in RCS directory (write protected)
- Check-in your file: `ci`
- Check-out (read only): `co`
- Check-out (read/write): `co -l`
- Read the log: `rlog`
- Check differences: `rcsdiff`

Minimal RCS

- `mkdir RCS`
- `ci -l myprog.f90`
- Use Ctrl-D to end the log message

Why use CVS?

- RCS locks files, CVS merges changes. Locking can be a problem e.g. large files.
- CVS handles remote repositories.
- CVS has better idea of a system composed of many files.
- CVS better than RCS for multi-developer and/or large projects.

CVS overview

- Check-out a working copy: `cvs co torus`
- Status: `cvs status; cvs -n update`
- Differences: `cvs diff`
- Check-in: `cvs commit`
- Apply patch: `cvs update`

It was working earlier...

- Go back to known good version:
  `cvs -q update -D “2008-05-22”`
- Use bisection:
  ```
  cvs update -D “2008-05-26” Good
  cvs update -D “2008-05-28” Bad
  cvs update -D “2008-05-27” Bad
  ```
- Problem introduced on 26/5/2008

Graphical clients

- Many to choose from
- Will use TkCVS as an example: multi-platform, multi system (RCS, CVS, Subversion)
- http://www.twobarleycorns.net/tkcvs.html

More advanced CVS

- Branching and merging.
- Stable repository: develop, test, commit.
- Unstable repository with tagged stable versions.
- Setting up a repository - see Eric's guides.
- http://cvsbook.red-bean.com/cvsbook.html

CVS limitations

- Rename file loses history
- Doesn't version directories and links properly
- Can't work off-line easily
- Version number is per-file not per-revision
- Commits are not atomic
- http://svnbook.red-bean.com/

And finally ...

- Version control is really useful
- RCS, CVS, Subversion
- More sophisticated stuff is possible
- Feedback - what do people use?
- Enough talking?