CVS...

...what is it?

...why should I use it?

...how do I get started?
CVS, what is that?

“Concurrent Versions System”

- Version control for files
  → database that takes care of versions of files
- Securely working on group projects, sharing files
- Work on several machines
What goes into CVS?

- Essential text files of any kind
  - source code (C, C++, Python, Matlab, ...)
  - LaTeX documents
  - ASCII files of any other kind (e.g., notes)
- Binary files possible, but not as many advantages
  - version control still available
  - seeing differences not possible
  - main problem: storage space requirements on server side → subversion (successor of CVS) will solve this problem (and others). Coming soon.
  - make only limited usage of CVS for binary files (e.g., figures in LaTeX document are o.k., segmentation result that changes every other day is not o.k.)
Scenarios

● Work in group (e.g., pulmonary project)
  ⇒ MUST GO INTO CVS, NO EXCEPTIONS!
  Reasons: other group members need access

● Work on multiple machines
  ○ Home and lab
  ○ Multiple workplaces in lab

  → do not try to copy files by hand, sooner or later you will lose data. Let CVS do the work!
Advantages

- Easy and secure sharing of code
- Several people can work on same file simultaneously
- Possibility of
  - visualizing differences
  - backtracking
  - retrieving previous release
  - branching, merging
Rules

- Commit frequently (at least once a day if files changed) → CVS is a working tool, not an archive for finished projects
- If it is shared code and other people may want to use it immediately: do not commit if it is broken (does not compile)
- Do not bother about others seeing/criticizing/laughing at your code, they have better things to do!
Setting up a private CVS repository

⇒ Great for your personal documents, things you do not want to share with others

How to do it:

● Fileservers moritz.ecn.uiowa.edu can host private CVS repositories

● CVS root directory:
   :pserver:username@moritz.ecn.uiowa.edu:/home/username/.cvsroot

● Same password as your system password

● For details refer to “Open Source Development With CVS”, p. 12
Security – tunneling through SSH

Problem: CVS (pserver) uses clear text (password, data)

Solution: set \texttt{CVS\_RSH} in your \texttt{.bashrc} file:

\begin{verbatim}
export CVS_RSH=ssh
\end{verbatim}
Using CVS — Commandline

A typical work-session with CVS:

START

1. cd projectDirectory
2. cvs update
3. <edit existing files>
4. cvs update
5. <resolve conflicts if necessary>
6. cvs commit

END

<add new file "foo.cc">

(for binary files:
   cvs add -kb foo.bin)

See CEIG lab manual & “Open Source Development With CVS” book (see below).
Using CVS — GUs (1)
Using CVS — GUIs (2)

Linux:


Windows:


Macintosh:

“Open Source Development with CVS” by Karl Fogel

free download of pdf file:  
http://cvsbook.red-bean.com/

Especially read the chapter "A day with CVS" (about 30 pages)
URL Summary

CVS home
http://www.cvshome.org/

Commandline version for Microsoft Windows
http://www.cygwin.com/

GUIs:
http://cervisia.sourceforge.net/  (Linux)
http://www.wincvs.org/  (MSWin, Mac)
http://www.lincvs.org/  (Linux)
http://www.tortoisecvs.org/  (MSWin)

Documentation “Open Source Development with CVS”
http://cvsbook.red-bean.com/

Subversion (successor of CVS):
http://subversion.tigris.org/