

**EECS22L LAB DISCUSSION**  
**WEEK 3**

**Che-Wei Chang**

# CVS-SET ENVIRONMENT VARIABLE

- Set the environment variable CVSROOT

- `setenv CVSROOT :ext:account@machine:path`

example : set CVSROOT to /users/ugrad2/2013/winter/teamX/cvs\_rep  
> `setenv CVSROOT :ext:teamX@ladera.eecs.uci.edu:/users/ugrad2/2013/  
winter/teamX/cvs_rep`

- P.S. the words in **RED** is the information you have to provide.

In this example, **account** = teamX

**machine** = ladera.eecs.uci.edu

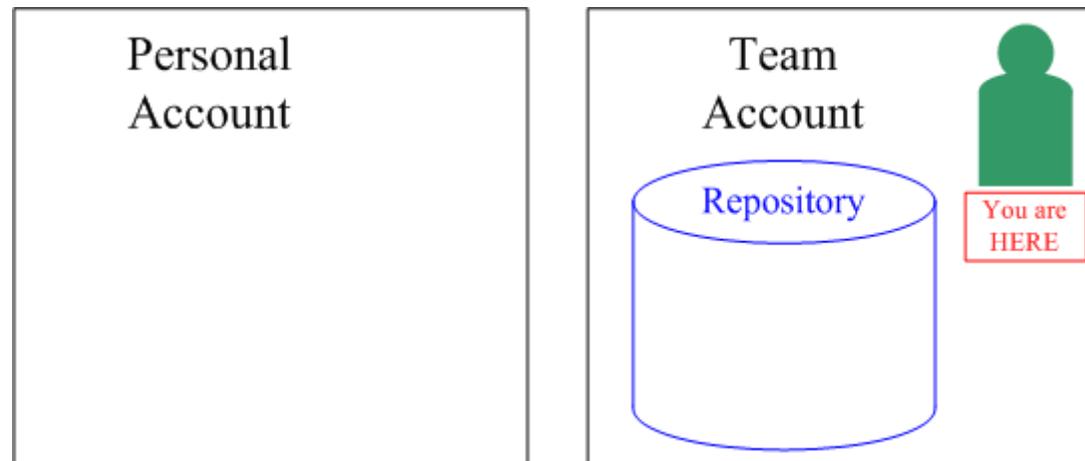
**path** = /users/ugrad2/2013/winter/teamX/cvs\_rep



# CVS-CREATE REPOSITORY

- Create a repository
  - `cvs -d ~directory init`

example : create repository at `/users/ugrad2/2013/winter/teamX/cvs_rep`  
> (login to teamX)  
> `mkdir cvs_rep` (if `cvs_rep` does not exist)  
> `cvs -d ~/cvs_rep init`



# CVS-IMPORT PROJECT

- Import project to repository

- `cvs import -m “message” project vender version`

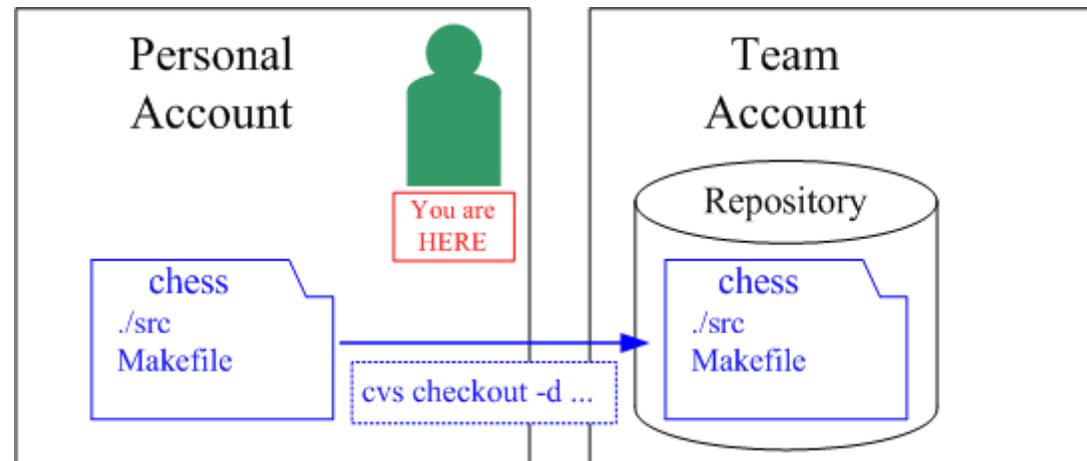
example : put project “chess” to repository

> (login to personal account and cd into your chess directory)

> (set environment variable CVSROOT if you have not done it yet. See page 2)

> `cvs import -m “chess alpha” chess chewei start`

P.S. after import the project into the repository you can delete your local directory, but better to do that after you confirm it is imported successfully



# CVS-CHECKOUT PROJECT

- Checkout a project from repository to local
  - `cvs checkout -d path project`

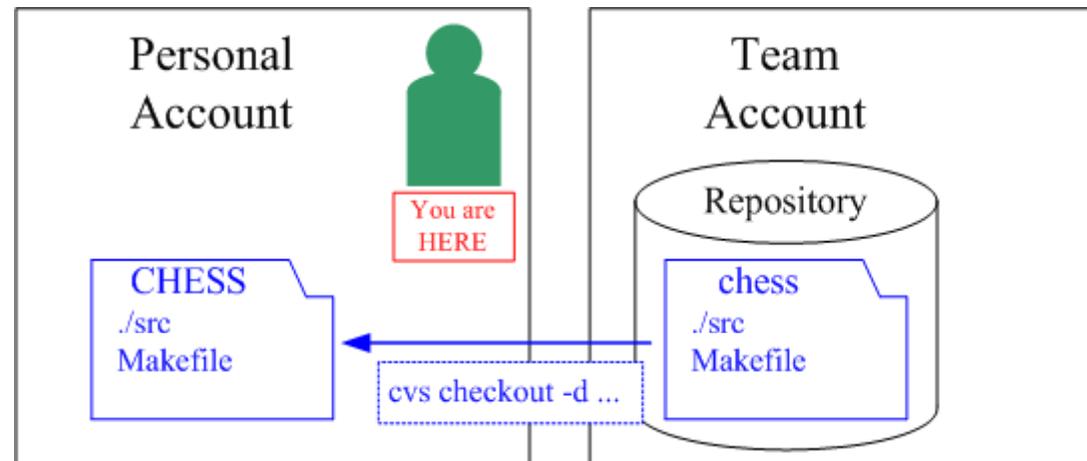
example : checkout project “chess” from repository to directory “CHESS”

> (login to your personal account)

> (set environment variable `CVSROOT` if you have not done it yet. See page 2)

> `cvs checkout -d CHESS chess`

P.S. `[-d path]` is just optional.



# CVS-ADD NEW DIRECTORY

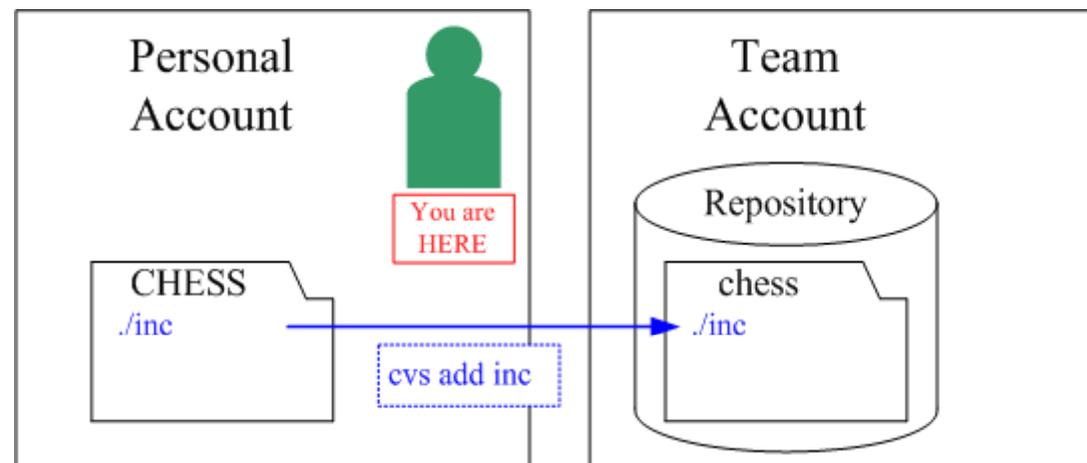
- Add a new empty directory to the repository
  - cvs add **directory**

example : add directory “inc” to the repository

> (login to your personal account and cd into your cvs checkout directory)

> (set environment variable CVSROOT if you have not done it yet. See page 2)

> cvs add inc

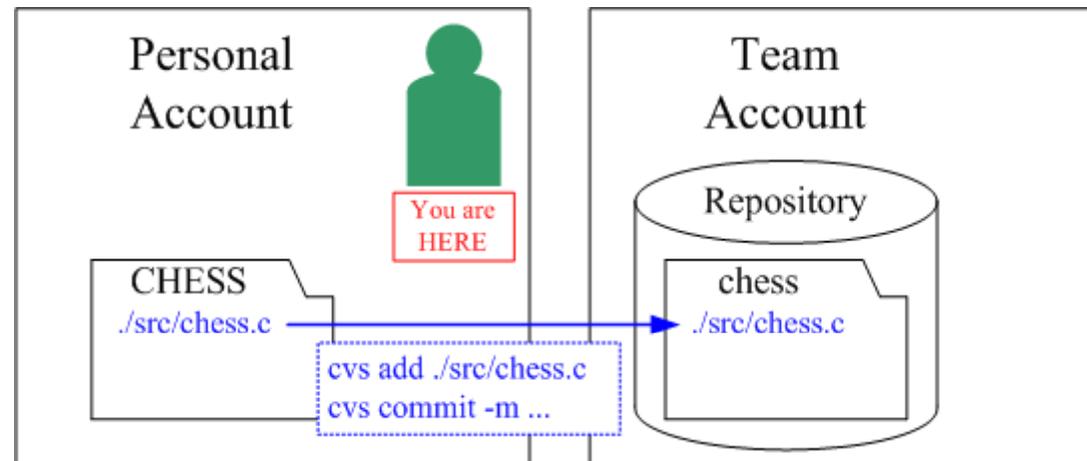


# CVS-ADD NEW FILE

- Add a new file to the repository
  - cvs add **file**  
cvs commit -m “**message**”

example : add file “./src/chess.c” to the repository

- > (login to your personal account and cd into your cvs checkout directory)
- > (set environment variable CVSROOT if you have not done it yet. See page 2)
- > cvs add ./src/chess.c
- > cvs commit -m “put chess.c in repository”



# CVS-UPDATE PROJECT

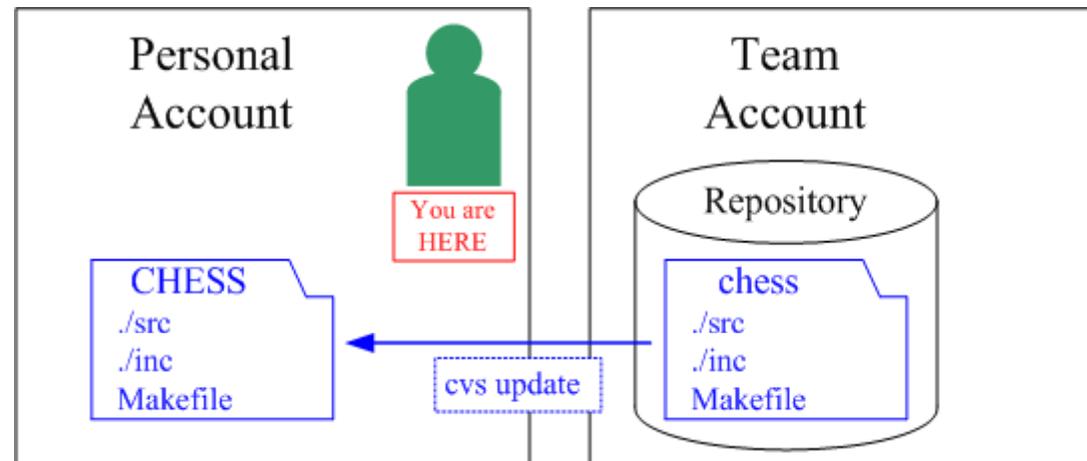
- Update your working directory
  - cvs update

example : update working directory “CHESS” from repository

> (login to your personal account and cd into your cvs checkout directory)

> (set environment variable CVSROOT if you have not done it yet. See page 2)

> cvs update



# CVS-COMMIT

- Commit changes to CVS repository
  - `cvcs commit -m "message" file`

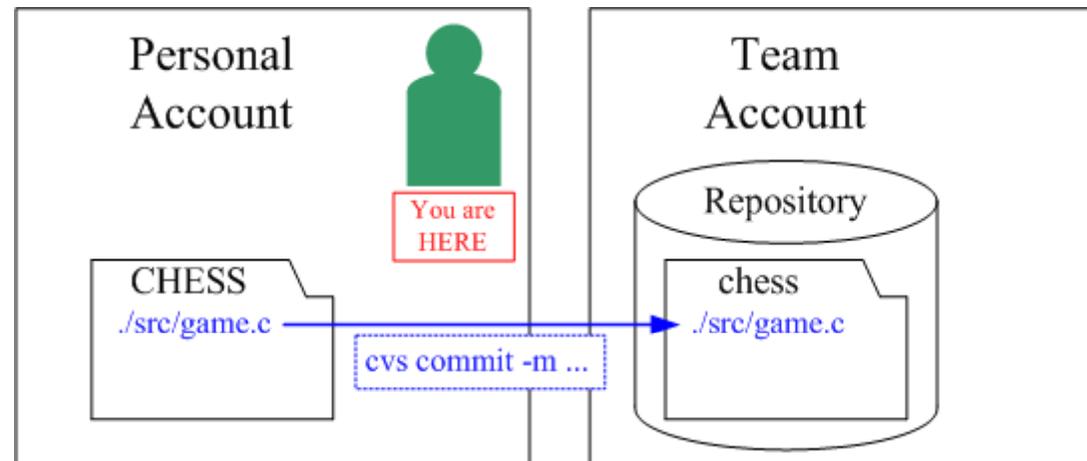
example : commit change in file `./src/game.c` to the repository

> (login to your personal account and cd into your cvs checkout directory)

> (set environment variable `CVSROOT` if you have not done it yet. See page 2)

> (modify `./src/game.c` and save the change)

> `cvcs commit -m "game.c modification" ./src/game.c`



# CVS-TAG PROJECT

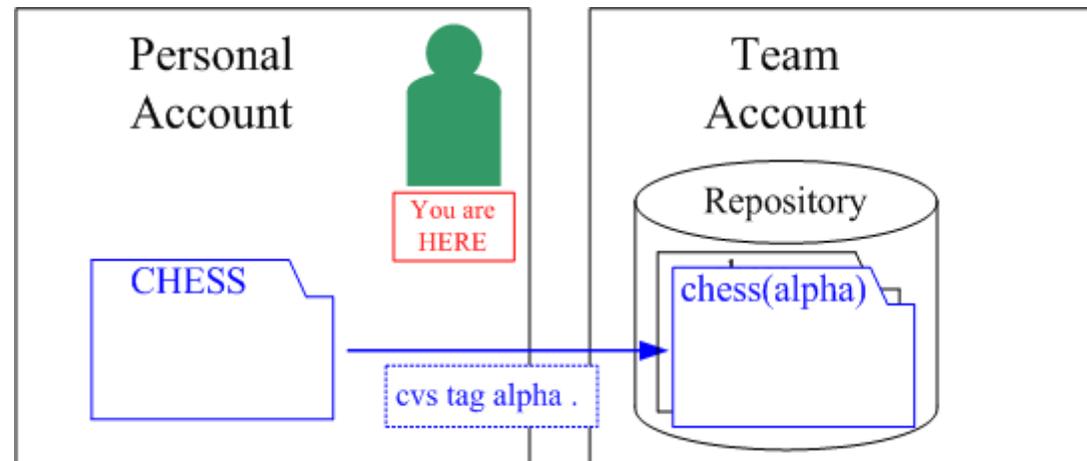
- Tag the project in CVS repository
  - cvs tag **version**

example : create branch “alpha” for your chess program

> (login to your personal account and cd into your cvs checkout directory)

> (set environment variable CVSROOT if you have not done it yet. See page 2)

> cvs tag alpha .



# CVS-CHECKOUT TAGGED PROJECT

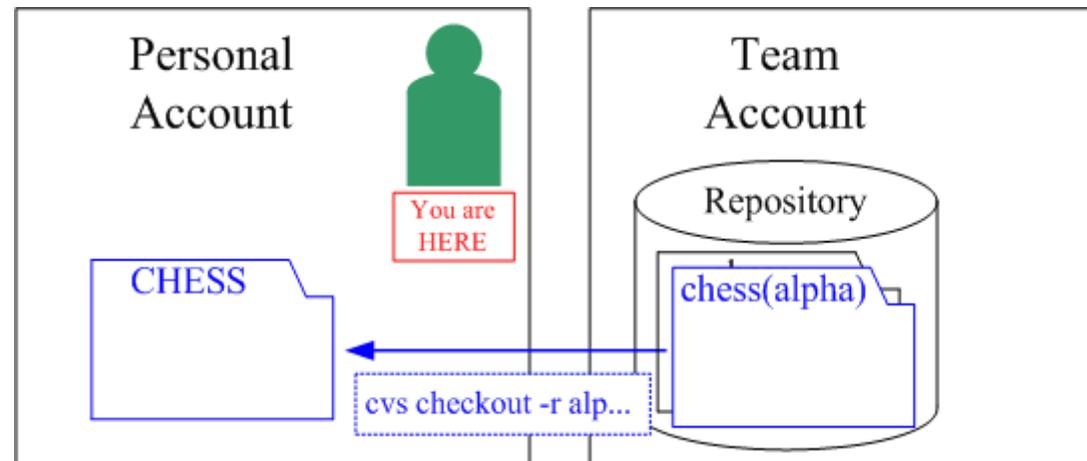
- Checkout a tagged project
  - `cvs checkout -r version project`

example : checkout branch “alpha” of chess from CVS

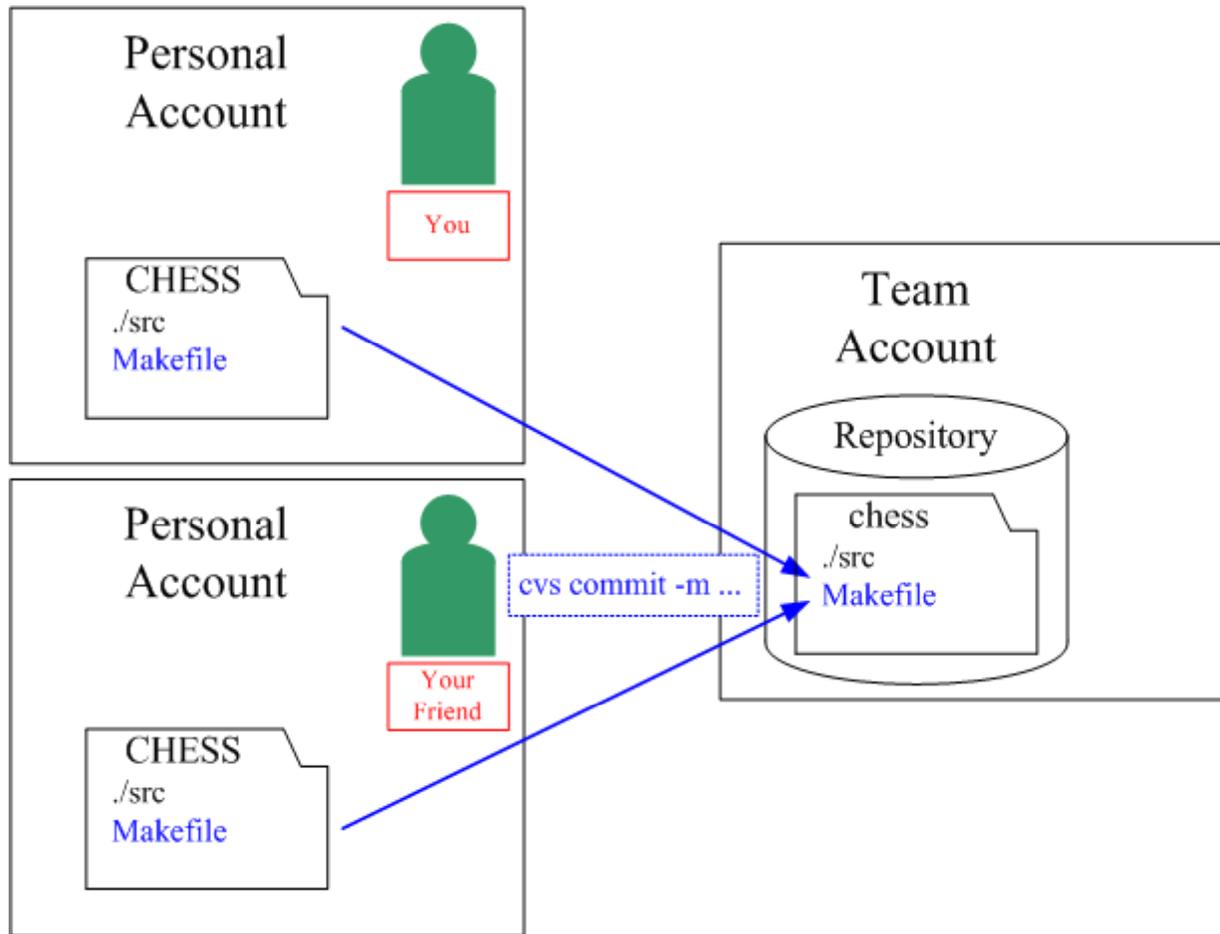
> (login to your personal account and cd into your cvs checkout directory)

> (set environment variable CVSROOT if you have not done it yet. See page 2)

> `cvs checkout -r alpha chess`



# CVS-RESOLVE CONFLICT



# CVS-RESOLVE CONFLICT (CONT)

- Resolve the conflict

example : you and your friend both modify Makefile and commit it to CVS.  
your friend committed it before you did.

> (you will not be able to commit the your change due to the conflict)

> cvs update ( C Makefile will show on the screen )

> vi Makefile

> (find the symbols inserted by CVS and remove it)

> (the symbols will be (<<<<<< filename, =====, and >>>>>> version)

> ( example :

<<<<<< Makefile ← Inserted by CVS

Your Code

=====

← Inserted by CVS

Your friend's Code

>>>>>> 1.2

← Inserted by CVS

)

> cvs commit -m "conflict resolved" Makefile



# GRADING CRITERIA FOR BETA VERSION

- What are we looking for in the beta version?
  - Complete and clean tarball
  - Proper project file hierarchy
  - README
  - ...
  - ...
  - ...
  - We will go to every team and explain the detail to you.  
The document will be posted on the website later.

