Subversion (svn) Basics

Department of Computer Science
Kent State University
Prof. Jonathan I. Maletic

Project Submission using svn

- Use svn for project and lab assignment submission
- svn is a tool for version control (git is another)
- Keeps a repository of what is under version control at a central location (department server)
- Allows users to make local (aka working) copies of the repository on their computer(s)
- Saves each version a user commits
- Administrator sets up main repository

Getting and Updating a Repo

- *Checkout* a repository. A local/working copy of a repository is created on a computer system. This links a given folder to the address of the repository.
  - `svn checkout` *address folder*
- *Update* a local/working copy. The contents of the repository are copied to the local folder. This updates the local copy.
  - `svn update`

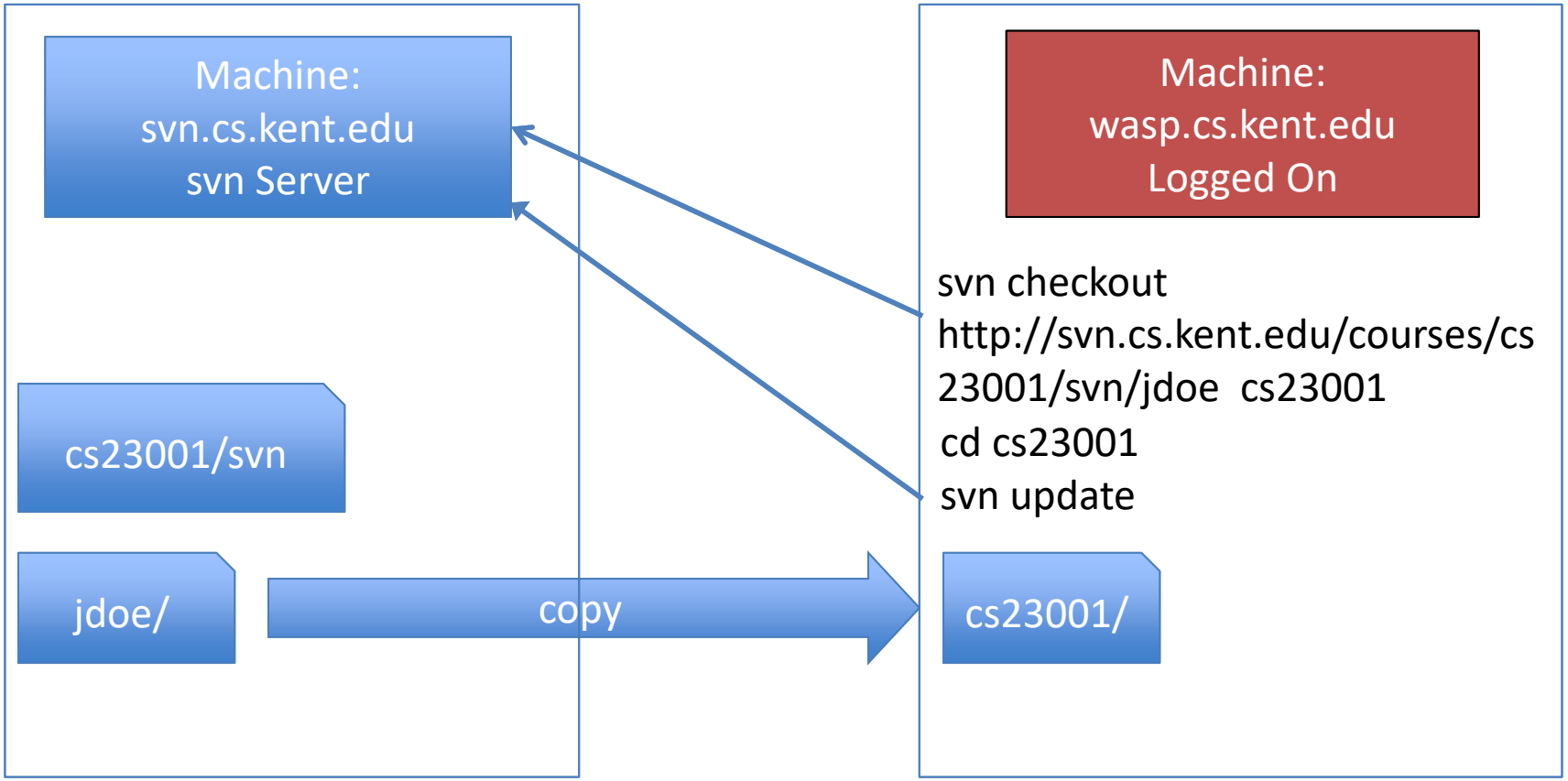
Committing Changes

- *Commit* a change. Copy a local change to repository. This "commits a change" to the repository. This is a new version of the item under version control. It is numbered and the old version is saved. Need to commit:
  - Modified files
  - Added or removed files or folders
  - `svn commit` *file* `-m` "*Commit note*"

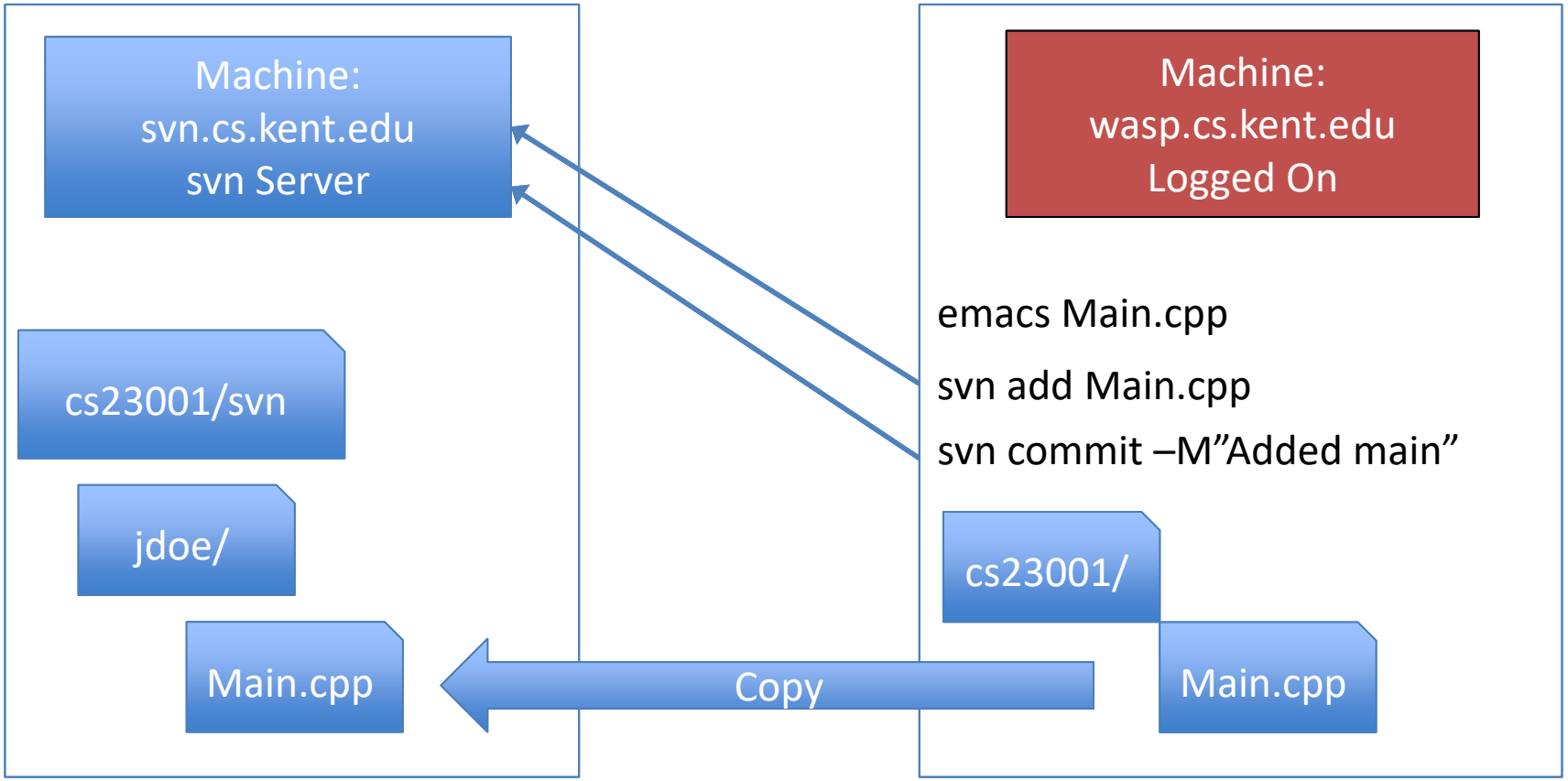
Other Essential Commands

- `svn add` – add a file to versioning system. A commit must be done after the add to commit it to the repository. Done once per file.
- `svn delete` – delete a file, followed by a commit.
- `svn log` – get all the commit messages. Use this to roll back to an earlier version.

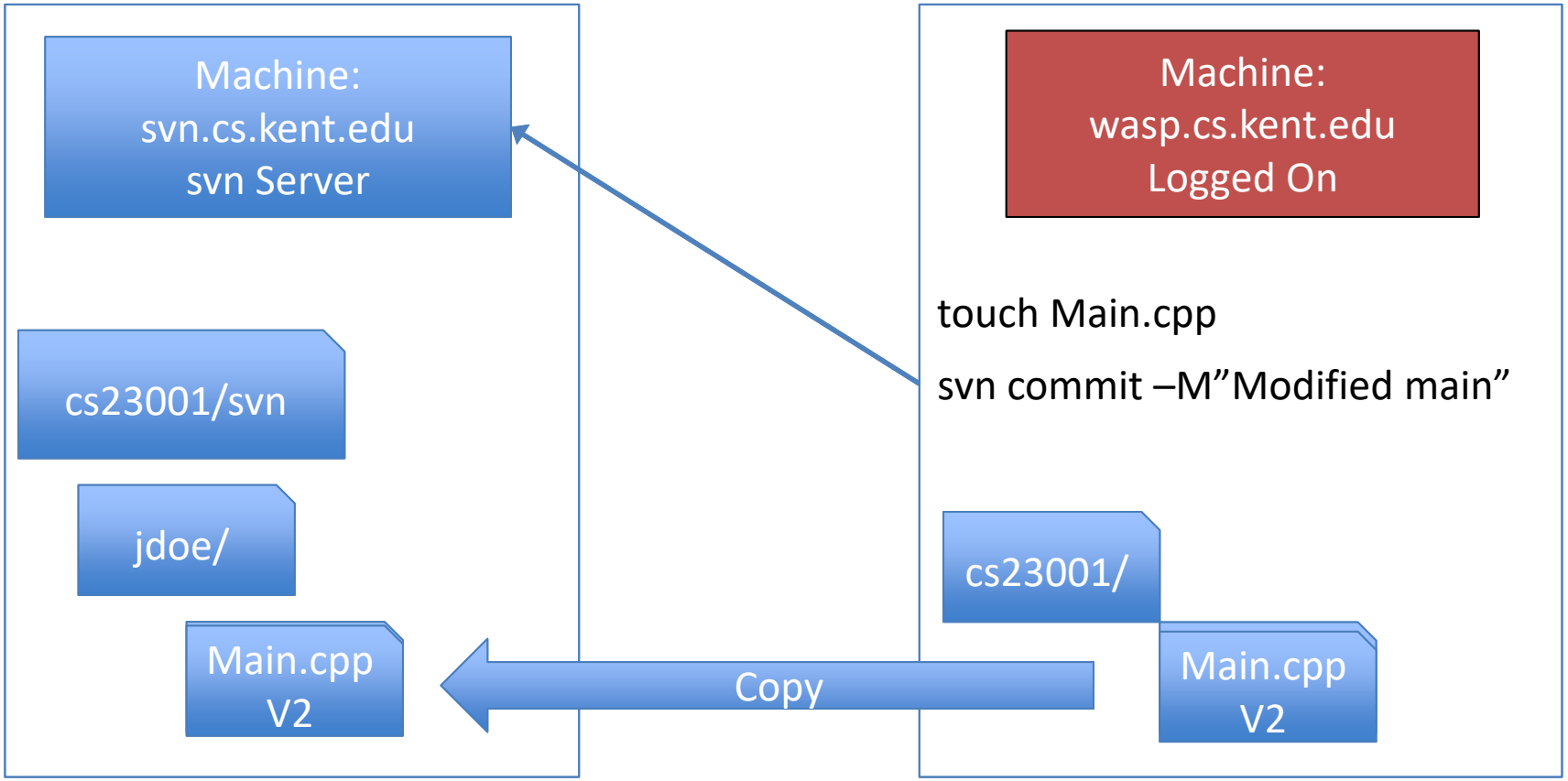
Check Out Working Copy



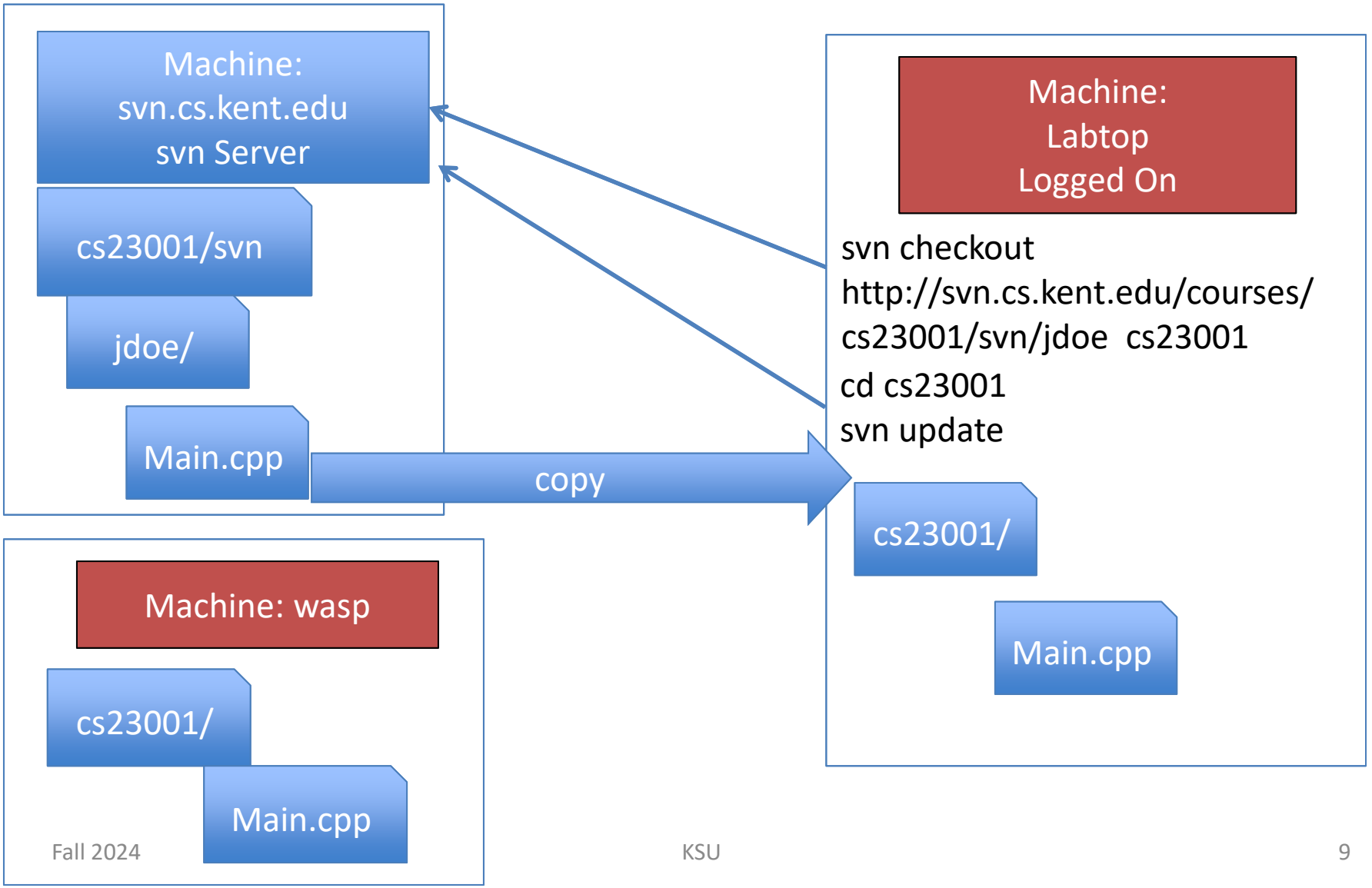
Add & Commit


Machine:
svn.cs.kent.edu
svn Server
Machine:
wasp.cs.kent.edu
Logged On
emacs Main.cpp
svn add Main.cpp
svn commit –M"Added main"
cs23001/svn
jdoe/
Main.cpp
Copy
cs23001/
Main.cpp

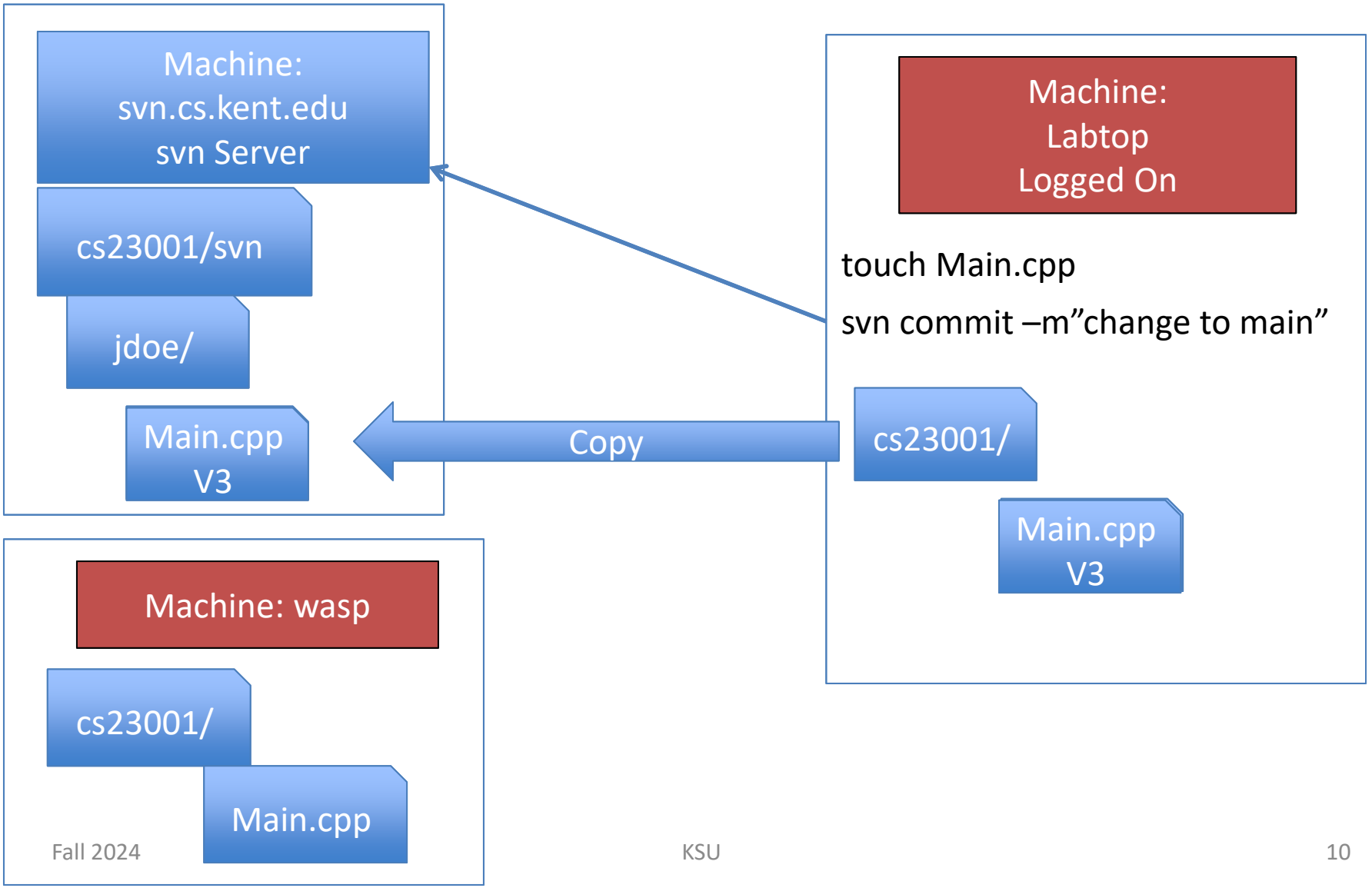
Commit a Change



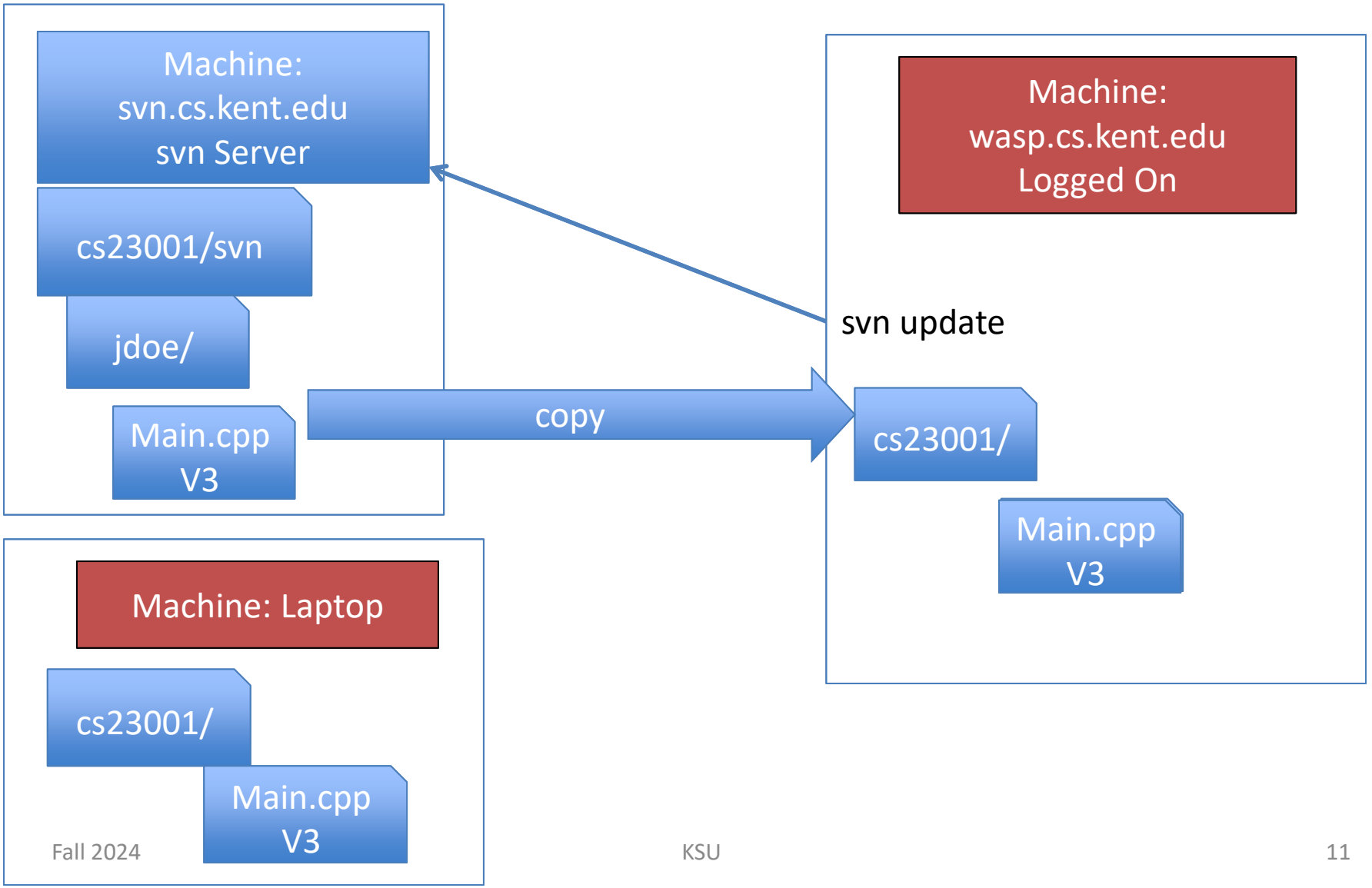
Multiple Working Copies



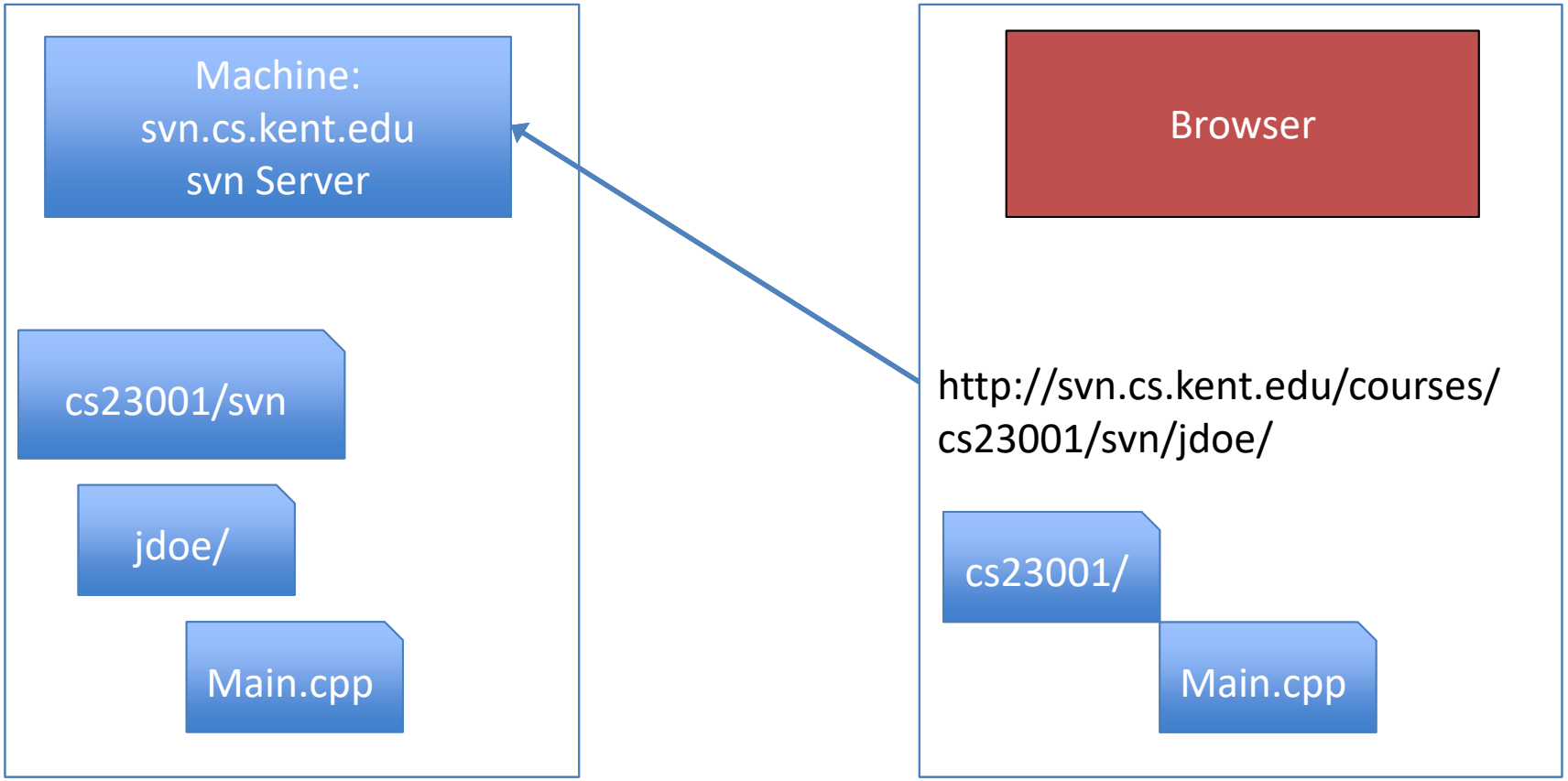
Commit a Change



Update wasp/hornet



Web Interface



Rules to Live By (svn)

- **Always** `update` before you start working
- **Always** `commit` after any change

- Commit often – after a major (or minor) change:
  - Add a method
  - Fix a bug
  - Change a method