

Warmup

A programmer's wife tells him, "Would you mind going to the store and picking up a loaf of bread? Also, if they have eggs, get a dozen."

The programmer returns with 12 loaves of bread.

Section 2:

Developer tools and you

Alex Mariakakis

cse331-staff@cs.washington.edu (staff-wide)

Outline

- Introductions
- SSH clients – PuTTY
- UNIX
- Source control – SVN
- Demo + questions

What is an SSH client?

- Uses the secure shell protocol (SSH) to connect to a remote computer
 - Enables you to work on a lab machine from home
 - Similar to remote desktop
- Windows and Linux users: PuTTY and WinSCP
 - Windows and Linux users
- Mac users: Terminal application
 - Go to Applications/Utilities/Terminal
 - Type in "ssh -XY cseNetID@attu.cs.washington.edu"

What is UNIX?

- Command-line based operating system
 - Like Windows or Mac OS without the mouse

Command	What it does
pwd	p rints the name of the w orking d irectory
ls	l ists the files in a directory (i.e., l ists s tuff)
cd	c hanges a d irectory
cp	c opies a file or directory
mv	m ove/rename a file or directory
rm	r emoves a file
mkdir	m ake a new d irectory
rmdir	r emove an empty d irectory
man	pulls up the m anual pages

DEMO #1

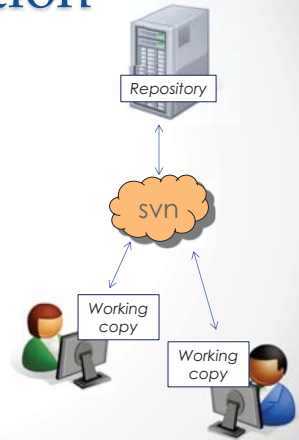
<http://courses.cs.washington.edu/courses/cse331/14wi/tools/WorkingAtHome.html>

What is source control?

- Also known as version control/revision control
- System for tracking changes to code
 - Software for developing software
- Essential for managing projects
 - See a history of changes
 - Revert back to an older version
 - Back up your work
 - Merge changes from multiple sources
- We'll be talking about Subversion, but there are alternatives
 - ✓ Git, Mercurial, CVS
 - × Email, Dropbox, USB sticks

Source control organization

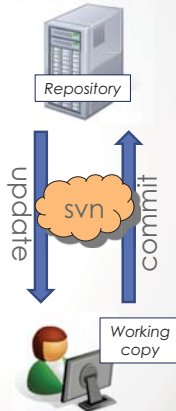
- A **repository** stores the master copy of the project
 - Someone creates the repo for a new project
 - Then nobody touches this copy directly
 - Lives on a server everyone can access
- Each person **checks out** her own **working copy**
 - Makes a local copy of the repo
 - You'll always work off of this copy
 - The version control system syncs the repo and working copy (with your help)



Source control common actions

Most common commands:

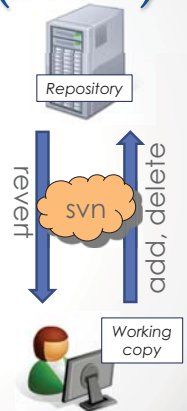
- **Commit / checkin**
 - integrate changes *from* your working copy *into* the repository
- **Update**
 - integrate changes *into* your working copy *from* the repository



Source control common actions (cont.)

More common commands:

- **Add, delete**
 - add or delete a file in the repository
 - just putting a new file in your working copy does not add it to the repo!
- **Revert**
 - wipe out your local changes to a file
- **Resolve, diff, merge**
 - handle a conflict – two users editing the same code



How to use Subversion

1. Eclipse plugin: Subclipse
2. GUI interface: TortoiseSVN, NautilusSVN
3. Command line: PuTTY

This Quarter

- We distribute starter code by adding it to your **repo**
- You will **code** in Eclipse just as you would have in your previous classes
- You turn in your files by **adding** them to the repo and **committing** your changes
- You will **validate** your homework by **SSHing** onto attu and running an Ant build file

DEMO #2

<http://www.cs.washington.edu/education/courses/cse331/14wi/tools/versioncontrol.html>

Theoretical Scenario

You are working on a computer in the lab and feel like you are at a good stopping point. You hit "Save" on your computer and start driving to Canada, only to realize that you forgot to commit your changes to your repo. Do you have to turn around and drive back?

Theoretical Scenario

You are working on a computer in the lab and feel like you are at a good stopping point. You hit "Save" on your computer and start driving to Canada, only to realize that you forgot to commit your changes to your repo. Do you have to turn around and drive back?

SSH into the attu, change to your eclipse workspace directory, and then call "svn commit"