CS 580 Client-Server Programming
Fall Semester, 2012
Doc 1 Introduction
Aug 28, 2012

Copyright ©, All rights reserved. 2012 SDSU & Roger Whitney, 5500 Campanile Drive, San Diego, CA 92182-7700 USA. OpenContent (http://www.opencontent.org/opl.shtml) license defines the copyright on this document.
Mercurial: The Definitive Guide, Bryan O'Sullivan,
http://hgbook.red-bean.com/read/index.html
Chapters 2, 3, 5.
Crashing

Last Day to Drop
Feb 2

Last Day to Add
Feb 4
Crashing Rules

As seats open up will fill them

Priority based on number of SDSU units in CS currently on your transcript
   Will alternate between undergrad and grad students

Give me a copy of unofficial transcript (email is fine)
   Need transcript at least 1 hour before class

People attending class have priority

Open University students have lower priority than SDSU students

Will start adding people Thursday in class
<table>
<thead>
<tr>
<th>Course Web Site</th>
</tr>
</thead>
<tbody>
<tr>
<td><a href="http://www.eli.sdsu.edu/index.html">http://www.eli.sdsu.edu/index.html</a></td>
</tr>
<tr>
<td>CS 580 Spring 12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Lecture Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Assignments</td>
</tr>
<tr>
<td>Wiki</td>
</tr>
<tr>
<td>Course Portal</td>
</tr>
<tr>
<td>Syllabus</td>
</tr>
<tr>
<td>Reading Assignments</td>
</tr>
</tbody>
</table>
Languages

Java
Client-Server

Client
- Initiates peer-to-peer communication
- Translate user requests into requests for data from server via protocol
- GUI often used to interact with user

Server
- Program that waits for incoming communication requests from a client
- Extracts requested information from data and return to client
What you will do

Implement Android client to talk to existing server using classic sock programming

Implement server for your existing client

Modify Android client to use HTTP to talk to existing server

Modify Android client to use distributed objects
Android

Use Eclipse to develop
   PC, Macs, Linux

Free download

Contains emulator but devices are better

Plan to spend about 3 weeks covering Android
   So will not cover all of Android
What makes Client-Server programming hard

Separate components

Client → Network → Server → Database
What makes Client-Server programming hard

Multitasking

Both client & server typically use threads to handle multiple tasks at the same time.
What makes Client-Server programming hard

Range of task you need to handle

- Network communication
- Threads
- UI design and implementation
- Database
- Protocol design
- Data security
- Authentication
- Authorization
Required of a Programmer

Designing robust protocols
Network programming
Designing usable computer-human interfaces
Good documentation skills
Good debugging skills
Understand the information flow of the company/customer
Mastery of concurrency
Multi-platform development
Database programming
Security
Scale Changes Everything

Sharpen your programming skills
# Names

<table>
<thead>
<tr>
<th></th>
<th>Java</th>
<th>Smalltalk</th>
<th>C#</th>
<th>Ruby</th>
</tr>
</thead>
<tbody>
<tr>
<td>Class</td>
<td>PascalCase</td>
<td>PascalCase</td>
<td>PascalCase</td>
<td>PascalCase</td>
</tr>
<tr>
<td>Method</td>
<td>camelCase</td>
<td>camelCase</td>
<td>PascalCase</td>
<td>foo_bar</td>
</tr>
<tr>
<td>Field</td>
<td>camelCase</td>
<td>camelCase</td>
<td>camelCase</td>
<td>@foo_bar</td>
</tr>
<tr>
<td>Parameter</td>
<td>camelCase</td>
<td>camelCase</td>
<td>camelCase</td>
<td>foo_bar</td>
</tr>
<tr>
<td>Local Variable</td>
<td>camelCase</td>
<td>camelCase</td>
<td>camelCase</td>
<td>foo_bar</td>
</tr>
</tbody>
</table>

Tuesday, August 28, 12
x = x + 1 //Add one to x
What does this do?

for i := 1 to n do
    MeetsCriteria[ i ] := True;
for i := 1 to n / 2 do begin
    j := i + i;
    while ( j <= n ) do begin
        MeetsCriteria[ j ] := False;
        j := j + i;
    end;
for i := 1 to n do
    if MeetsCriteria[ i ] then
        writeln( i, ' meets criteria ' );
What does this do?

for PrimeCandidate:= 1 to Num do
  IsPrime[ PrimeCandidate ] := True;

for Factor:= 1 to Num / 2 do begin
  FactorableNumber := Factor + Factor ;
  while ( FactorableNumber <= Num ) do begin
    IsPrime[ FactorableNumber ] := False;
    FactorableNumber := FactorableNumber + Factor ;
  end;
end;

for PrimeCandidate:= 1 to Num do
  if IsPrime[ PrimeCandidate] then
    writeln( PrimeCandidate, ' is Prime' );
Source Control

Track changes in software

Maintain software in one location

Multiple people can make updates
Common Free Source Control Systems

CVS
Concurrent Versions System
Command line interface in Unix
Various interfaces in Window

Subversion
Claims to be a better CVS
Many commands are same as CVS

Git
Created by Linus Torvald
Distributed Version control

Mercurial
Python based
Distributed version control
# Mercurial

http://mercurial.selenic.com/

<table>
<thead>
<tr>
<th>Runs on:</th>
<th>Command line interface</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mac OS X</td>
<td>GUI interface for windows</td>
</tr>
<tr>
<td>Unix</td>
<td>Eclipse plugins</td>
</tr>
<tr>
<td>Linux</td>
<td>Can use locally with no server</td>
</tr>
<tr>
<td>Windows</td>
<td></td>
</tr>
</tbody>
</table>
Mercurial in Eclipse

Simple Workflow

Start repository

Commit changes

Code

Commit changes (creates changeset)

Commit changes (creates changeset)

Code

Code
Changesets

Revision Number
changeset: 4 : 2278160e78d4

tag: tip
user: Bryan O'Sullivan <bos@serpentine.com>
date: Sat Aug 16 22:16:53 2008 +0200
summary: Trim comments.
Branch in one Repository

- Start repository
- Code
- Commit changes
- Merge
- Commit changes
- Code
- Commit changes
- Code
- Start repository
Standard Mercurial Workflow

Commit changes

code

Start repository
Standard Mercurial Workflow

Commit changes

Start repository

code

hg clone

Commit changes

Start repository
code
Standard Mercurial Workflow

- Start repository
  - code
  - Commit changes

- Commit changes
  - code
  - Start repository

Commit changes
Standard Mercurial Workflow

Commit changes

code

Commit changes

code

Start repository

Commit changes

code

Commit changes

code

Start repository

hg pull

hg update

Tuesday, August 28, 12
Basic Source Control Operations

Starting a new project
Adding code to a project
Modifying existing code
Retrieving past versions of code
Handling conflicts in code
Creating code branches
Merging code branches

Cloning repositories
Pulling repositories
Pushing repositories
Mercurial Commands
Creating a Repository

hg init myproject

In Eclipse right click on a project and select the "Share Project..." item in the "Team" menu
Repository & Working Directory

Repository
In .hg directory
Contains
  history of changes
  list of files part of project

Working Directory
Contains
  Project files
  Project directories
  .hg
Adding Files to Repository

Must tell Repository which files to maintain

hg add
   Adds all files in the current directory

hg add filename
   Adds the named file

Just adds files to list of files to maintain
Does not add contents of files to repository
Committing files to Repository

hg commit
Adding/Committing in Eclipse

[Image showing the Eclipse IDE interface with options for adding and committing changes]

- New
- Go Into
- Open in New Window
- Open Type Hierarchy
- Show In
- Copy
- Copy Qualified Name
- Paste
- Delete
- Remove from Context
- Build Path
- Source
- Refactor
- Import...
- Export...
- Refresh
- Close Project
- Close Unrelated Projects
- Assign Working Sets...
- Run As
- Debug As
- Validate
- Team

[Image showing a commit dialog in Eclipse]

- Commit...
- Push...
- Pull...
- Update
- Switch To...
- Apply Patch...
- Import Patch...
- Export Patch...
- Tags...
- Bookmarks...
- Branch...
- Merge...
- Rebase...

- Sample Commit
- Select old commit message
- User to record as committer: whitney
- Select Files:
  - File: .classpath
  - Status: Untracked
  - File: project
  - Status: Untracked
  - File: .settings/org.eclipse.jdt.core.prefs
  - Status: Untracked
  - File: src/edu/sdsu/cs/cs5780/Example.java
  - Status: Untracked

- Select/unselect all
- Show added/removed files
- Revert unchecked resources

[Additional options and buttons for committing changes]
Reverting to different Version

Switch to...
Please enter a valid revision (local, global, tag or branch):
3:Sec9f6625587833f9b99753f422c2589c489d69

<table>
<thead>
<tr>
<th>Rev</th>
<th>Global</th>
<th>Date</th>
<th>Author</th>
<th>Branch</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>9a223dee086bf785425f90</td>
<td>2010-01-26 10:47 -0800</td>
<td>whitney</td>
<td></td>
<td>added a comment</td>
</tr>
<tr>
<td>4</td>
<td>67dbab268f18990da3cc3</td>
<td>2010-01-26 10:45 -0800</td>
<td>whitney</td>
<td></td>
<td>Just some code for an update</td>
</tr>
<tr>
<td>3</td>
<td>Sec9f6625587833f9b99753</td>
<td>2010-01-26 10:42 -0800</td>
<td>whitney</td>
<td></td>
<td>Something</td>
</tr>
<tr>
<td>2</td>
<td>5ff016cfff3935668fc4fc51</td>
<td>2010-01-26 09:47 -0800</td>
<td>whitney</td>
<td></td>
<td>Add test class</td>
</tr>
<tr>
<td>1</td>
<td>735c23eca0c50a3fa357be5</td>
<td>2010-01-26 09:27 -0800</td>
<td>whitney</td>
<td></td>
<td>Implemented adder</td>
</tr>
<tr>
<td>0</td>
<td>13e89dc77a9077f2d09268</td>
<td>2010-01-26 09:25 -0800</td>
<td>whitney</td>
<td></td>
<td>Initial commit</td>
</tr>
</tbody>
</table>
Creating Branches
Merging - without conflicts

use the merge item in the team menu and commit
Uploading to BitBucket

After creating a project in BitBucket
Use the Push item in the team menu
Downloading From Bitbucket

Clone repository
Create a clone from another repository.

Repository location
URL https://rogerwhitney@bitbucket.org/rogerwhitney/junitexample/ Local...

Authentication
Username rogerwhitney
Password ************

Clone destination
Parent directory of clone (default: workspace)
Clone directory name (default: basename) junitExample
Search for .project files in clone and use them to create projects.

Options
Abort clone when a timeout occurs
Do not update the new working directory
Use pull protocol to copy metadata
Use uncompressed transfer (fast over LAN)
Changeset you would like to have after cloning

42