Teaching Operating Systems Using Code Review

Christoffer Dall and Jason Nieh
Columbia University
Learning OS is hard

- Operating Systems are large and complicated
- Hands-on experience is crucial
- Modification to large and unfamiliar code bases
Teaching OS is hard

• Programming assignments must be reviewed
• Code is dense and spread out
• Disproportionate amount of time spent evaluating compared to teaching
GradeBoard

- Code review system designed for the classroom
- Web-based, intuitive, easy-to-use
- Build on Git and ReviewBoard
- Supports entire workflow of teaching OS
Outline

• Usage Model
• System Architecture
• Evaluation
• Conclusions
Usage Model

• Instructors and students

• Instructors *create, distribute*, and *grade* programming projects

• Students *download, backup, collaborate*, and *submit* programming projects
The Web Interface

<table>
<thead>
<tr>
<th>Starred Reviews</th>
<th>Outgoing Reviews</th>
<th>Incoming Reviews</th>
<th>All My Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>10</td>
<td>829</td>
<td>11</td>
</tr>
</tbody>
</table>

**All Incoming Review Requests**

<table>
<thead>
<tr>
<th>Summary</th>
<th>Submitter</th>
<th>Reviews</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homework 2 group submission (team10)</td>
<td>team10</td>
<td>4</td>
</tr>
<tr>
<td>Homework 2 group submission (team11)</td>
<td>team11</td>
<td>1</td>
</tr>
<tr>
<td>Homework 2 group submission (team12)</td>
<td>team12</td>
<td>1</td>
</tr>
<tr>
<td>Homework 2 group submission (team13)</td>
<td>team13</td>
<td>1</td>
</tr>
<tr>
<td>Homework 2 group submission (team14)</td>
<td>team14</td>
<td>3</td>
</tr>
<tr>
<td>Homework 2 group submission (team15)</td>
<td>team15</td>
<td>1</td>
</tr>
</tbody>
</table>
The Web Interface
Instructor Feedback

Grade Student A

Grade Student B

Grade Student C

Grade Student D

Release all grades

Time
You need to handle page_mapcount == 0 (think about the ZERO_PAGE).

Points deducted: 3

Team14
Do you mean we should have printed 1 in page_mapcount == 0? We felt it is wrong as user cannot find whether it is ZERO_PAGE or one page table entry pointing to the physical page. Could you clarify how we need to handle page_mapcount == 0 in different way without removing the meaning of each value mapcount can take?

Christoffer Dall
just check for mapcount == 0 and print 0, and if you know that at least the process printing has a reference, so 0 is not the case. The ZERO_PAGE is an optimization, and the mapping will be changed to a dedicated page once you write to that page, so the analogy is having a single dedicated page referenced.

The homework does not allow you to print 0, and my guess is that you did not even think about the fact that page_mapcount can be 0.

You also have no comment explaining this.

Team14
We discussed about ZERO_PAGE due to using the mm_struct during our demo. Forgot to include comment regarding that in code. Hence in the test case 2, to get 1's in all pages, we write to all pages we have mapped. However irrespective of this, printing 1 in case mapcount == 0 is wrong. The process that is printing has only read reference. Multiple process/threads could
System Architecture

Git

ReviewBoard

Management
Review Board

• Web-based interface

• Commercially maintained open-source tool

• One trivial change to the authentication system to prevent students from modifying their submissions
Management Tool

• Abstracts away low-level commands to classroom commands like:
  • create-homework
  • distribute-homework
  • upload-reviews
• Keeps single central list of registered students
OS @ Columbia

- 6 two-week programming assignments
- Modifying the Linux kernel for Android
- Intro to C, system calls, synchronization, virtual memory, scheduler, file systems
- Live demonstrations
- Careful code review
Evaluation

• Used GradeBoard at Fall 2011 Columbia OS course

• Over 100 registered students

• Survey participation: 57%
GradeBoard Improved Homework Understanding

- No: 8%
- Yes: 92%
Preferred compared to e-mailed score sheets

- No: 19%
- Yes: 81%
Preferred compared to comments in Git

81% Yes
19% No
Encouraged well-formatted code

46% No

54% Yes
“I wish I had this tool when I taught operating systems last year.”

—Nicolas Viennot, Head TA Fall 2009
Conclusions

- Implemented and evaluated GradeBoard

- Students learn better with GradeBoard for 2 reasons:
  - Comments inlined with student work
  - Student feedback identifies problem areas

- Over 80% of all submitted projects were discussed

- Free and maintained

- Students and instructors prefer GradeBoard over known alternatives
Questions?
Backup Slides...
Potential Solutions

• Grade Sheets: Manual and tedious

• Inline comments: requires use of command-line tools, time consuming

• Difficult for students to use
Problem Space

• Most hands-on OS assignments are based on some existing code base

• Both teaching OSes and commercial OSes

• No read-from-start-to-finish
Problem Space: Options

- Live demonstrations
- Automated Testing
- Doesn’t work, must evaluate the code