

What Should a Systems Administration Student's Homework Look Like?

LCA 2015

Tom Clark

Otago Polytechnic
Dunedin, New Zealand

Where to begin?

My mouth tends to run ahead of my brain at times...

Teaching systems administration is still fairly new.

- We've been teaching programming for a long time. We know what to teach and how to teach it¹.
- What instruction there is tends to be vendor/certification driven.
- We need to identify some durable core principles to share with students.
- Students need to learn *why* we do things, not just how to do them.

¹This is a lie. We don't really know these things.

Now, how am I going to pull this off?

I reached out to my network of friends and colleagues in systems administration.

- They confirmed that there was a need for sysadmin graduates.
- They were interested in students who had run multi-tier, multi-server services.

I mixed in a few of my teaching principles.

- Doing things is better than talking about doing things.
- Consistent thread: Things we do on the first day should be relevant on the last day.
- “Realism”:
 - Real tasks
 - Real tools
 - **Real assessment**

We've got 16 weeks. What will we do?

- Sysadmin professional practice.
 - Ticketing (RT)
 - Documentation (MediaWiki)
- Configuration management (Puppet)
- System monitoring (Nagios)
- Backup and recovery (Bacula)

Students, working in pairs, are given control of a set of virtual servers to manage.

What do we use that stuff for?

- Remember that we want to perform real tasks, so we won't just set up something like Puppet for its own sake.
- This leads to our *real assessment*: Students will deploy and operate a service (OwnCloud) using the tools they set up earlier in the semester.

How does the assessment work?

Student pairs are directed to deploy OwnCloud and keep in running around the clock over approximately two weeks. Half of their mark comes from OwnCloud uptime.

- They need to deploy on time.
- They need to resolve problems quickly.

How does the assessment work?

The remaining half of the the students' mark comes from their handling of issues raised in tickets and from troubleshooting problems.

- I play the role of an ordinary user or a manager by opening tickets.
- Students need to close tickets promptly.
- They need to document their work and communicate with affected parties.

What sorts of tickets?

- Ordinary user: password reset, restore a deleted file
- IT Manager: bulk addition of users, new Nagios check, configuration change

So far, this sounds kind of easy...

To make reaching their uptime goals a bit more challenging, I just deliberately break stuff. I've got root on all of these servers².

- Sometimes I just quietly shut down MySQL.
- At some point I do something brutal, like deleting important files and defacing the web site. For optimal learning experience, I like to do this at about 3:00 AM.

²One cunning student team set up a Nagios check that triggered an alert whenever I logged in to a server.

How do I observe student performance?

- I monitor uptime with my own instance of Nagios.
- I look at RT ticket logs and their wiki to see how they handled issues and worked as a team.

After the assessment period, I have a face-to-face postmortem session with each team.

So, how did it go?

- Uptime: surprisingly well!
- Resolving issues: Most issues were resolved very promptly. Many teams didn't make good use of RT to note how they solved problems.
- Other documentation: Look, none of us are perfect.

Some other notes on student performance

- Most students aren't entering the class with as much Linux proficiency as I'd like.
- Students set up a (closed) Facebook group to share information. This is mostly a good thing, but it requires me to be a little more unpredictable.
- Students enjoy the class. Students already in work said that they applied lessons learned in the class in their workplaces.

Future work

- I need to build or find a software tool to automate the breaking of student systems - sort of a configuration antimanager.
- I am (continually) in the process of putting class resources online.

Questions or feedback

@tom_clark

<https://github.com/tclark/op-papers>