



myExperience Report

Semester 2, 2018

Faculty: Faculty of Engineering

School: School of Computer Sci & Eng

Course: COMP9242 Advanced Operating Systems

Evaluation period: Oct 8 2018 12:00AM - Nov 1 2018 12:00AM

Course Report

Response Data

Raters	Student
Responded	22
Invited	31
Response Ratio	71.0%

Comparison of results for "Overall I was satisfied with the quality of the course"

This course: COMP9242 Advanced Operating Systems

1. Overall I was satisfied with the quality of the course				
Options	Count	Percentage	Statistics	Value
Strongly disagree	0	0.0%	Mean	5.55
Disagree	0	0.0%	Median	6.00
Moderately disagree	0	0.0%	Standard Deviation	0.60
Moderately agree	1	4.5%	Standard Error (base on SD)	0.13
Agree	8	36.4%	% Agreement	100.0%
Strongly agree	13	59.1%		

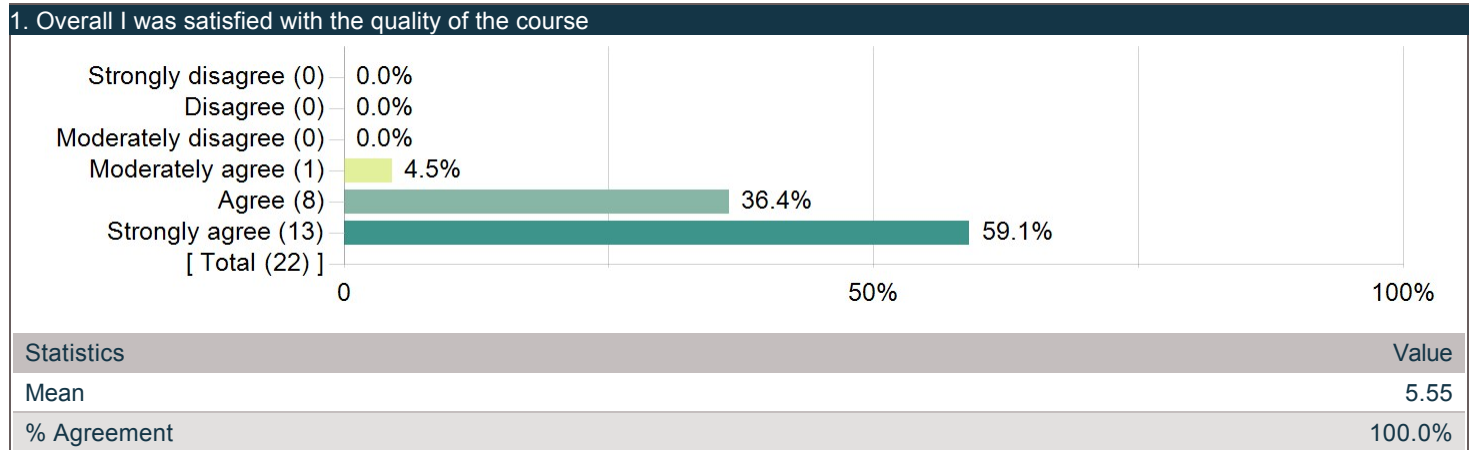
SCHOOL: School of Computer Sci & Eng

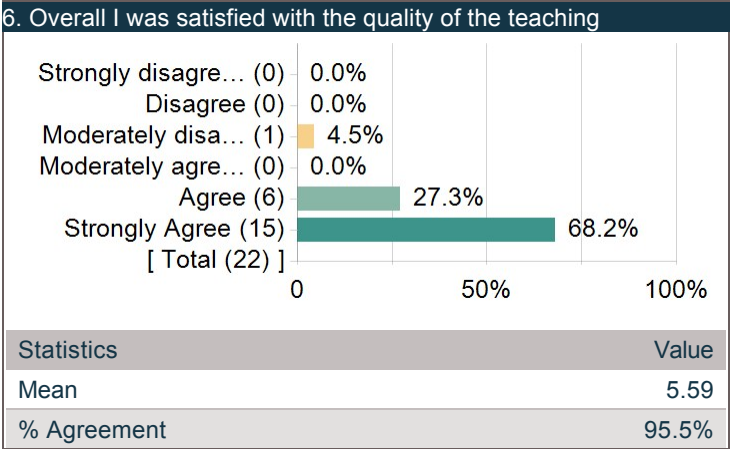
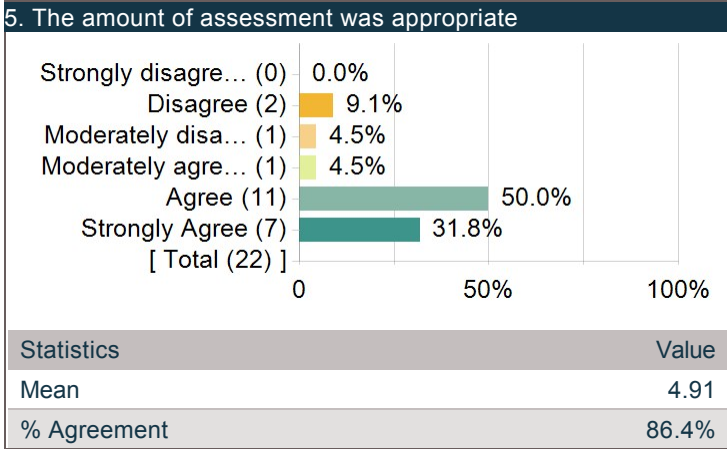
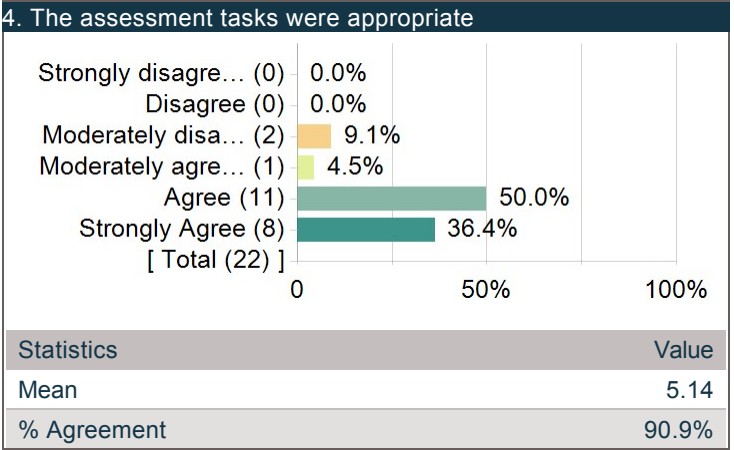
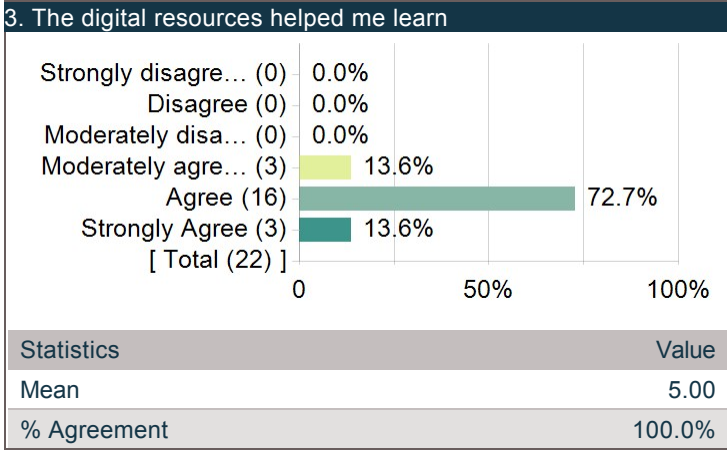
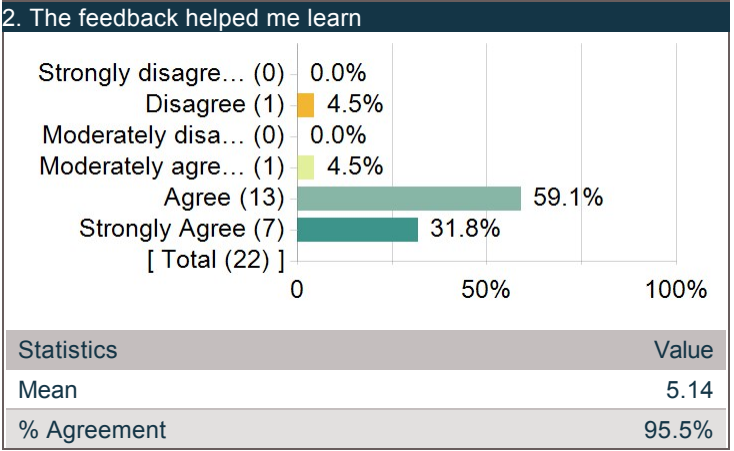
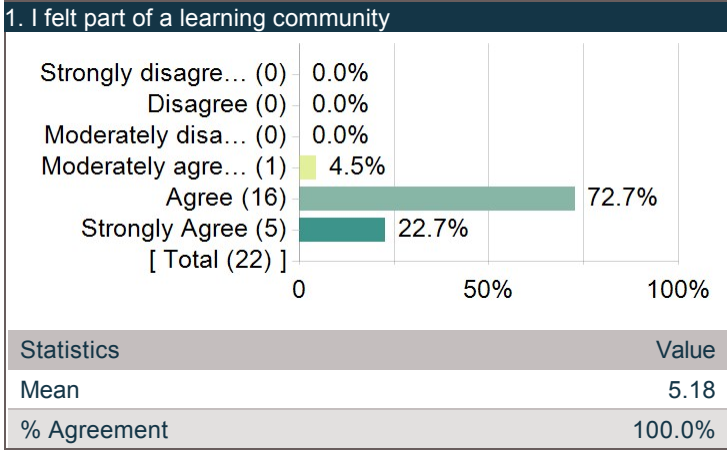
1. Overall I was satisfied with the quality of the course			
Options	Percentage	Statistics	Value
Strongly disagree	2.7%	Mean	4.81
Disagree	3.3%	Median	5.00
Moderately disagree	6.0%	Standard Deviation	1.18
Moderately agree	16.6%	Standard Error (base on SD)	0.02
Agree	40.8%	% Agreement	88.0%
Strongly agree	30.6%		

FACULTY: Faculty of Engineering

1. Overall I was satisfied with the quality of the course			
Options	Percentage	Statistics	Value
Strongly disagree	3.2%	Mean	4.81
Disagree	3.2%	Median	5.00
Moderately disagree	5.3%	Standard Deviation	1.20
Moderately agree	16.1%	Standard Error (base on SD)	0.01
Agree	41.8%	% Agreement	88.3%
Strongly agree	30.4%		

Overall I was satisfied with the quality of the course

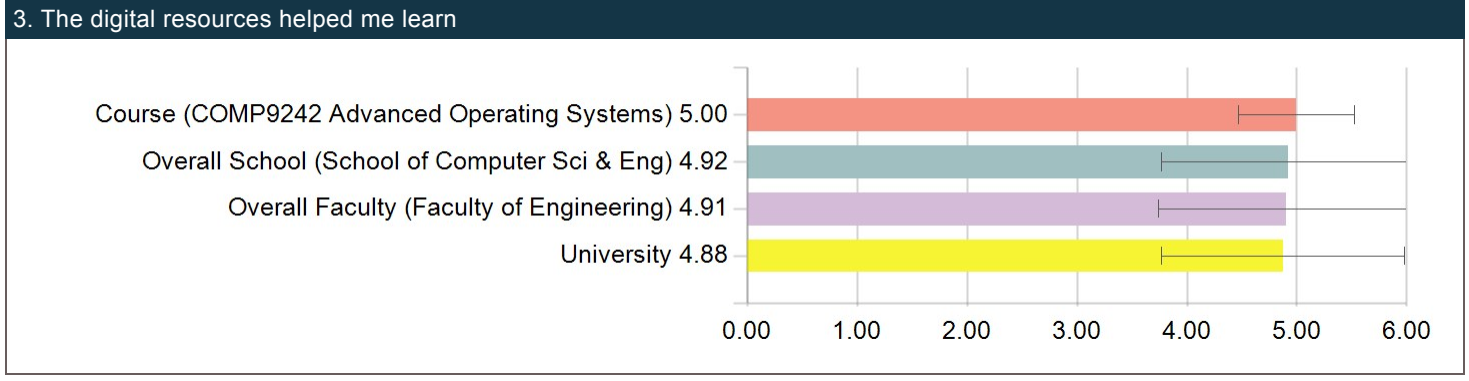
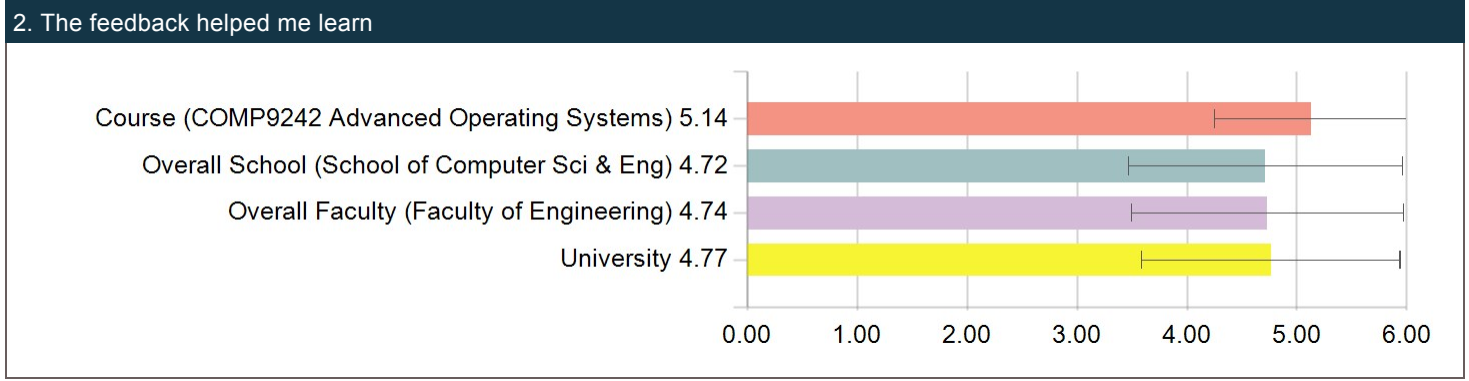
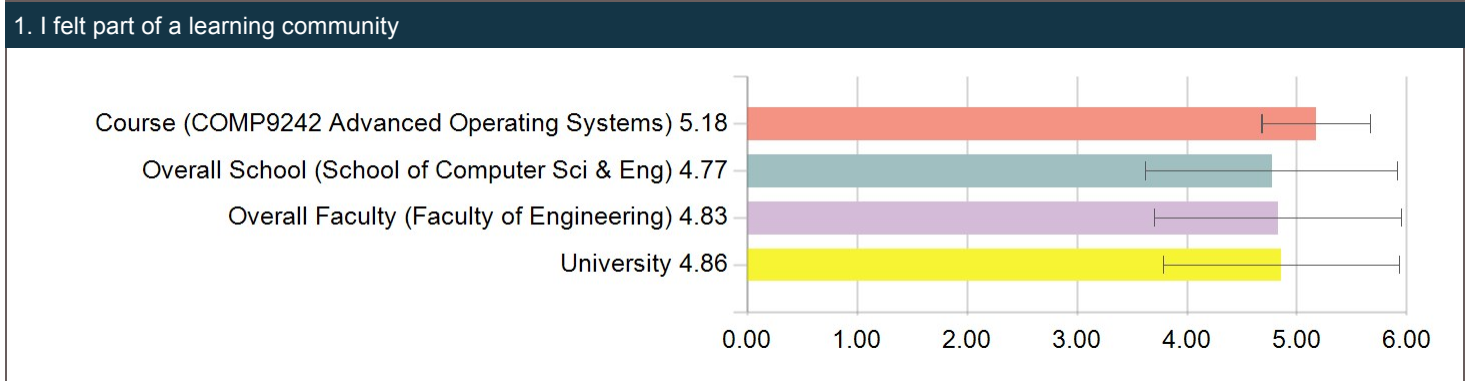
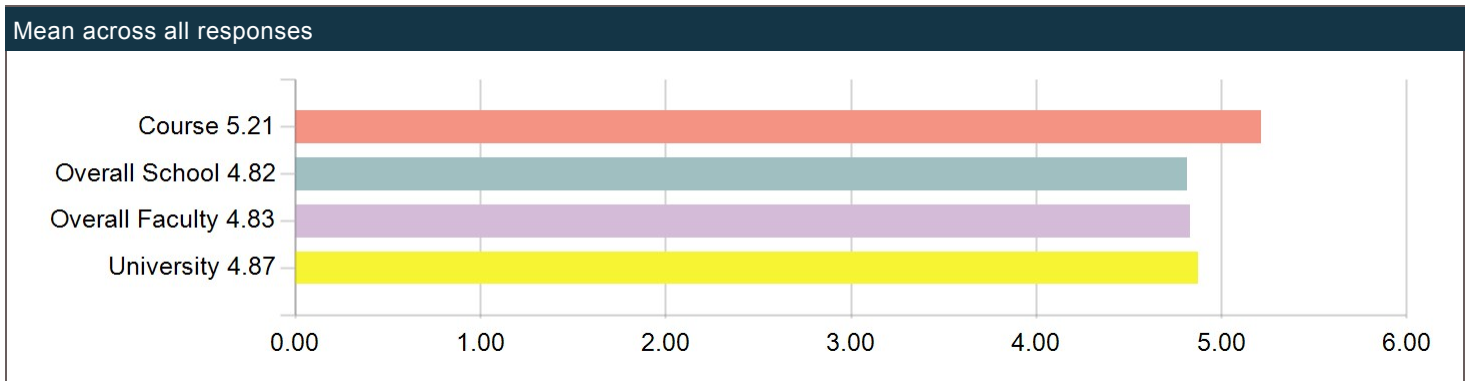




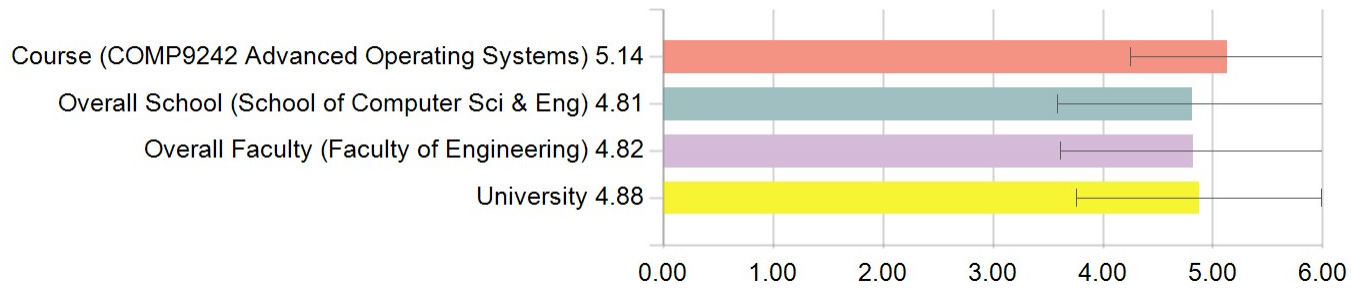
Comparison Statistics

Mean (average student responses between 1 and 6) and StandardDev (Standard deviation of student responses) are used for comparison statistics between Course, School, Faculty and University.

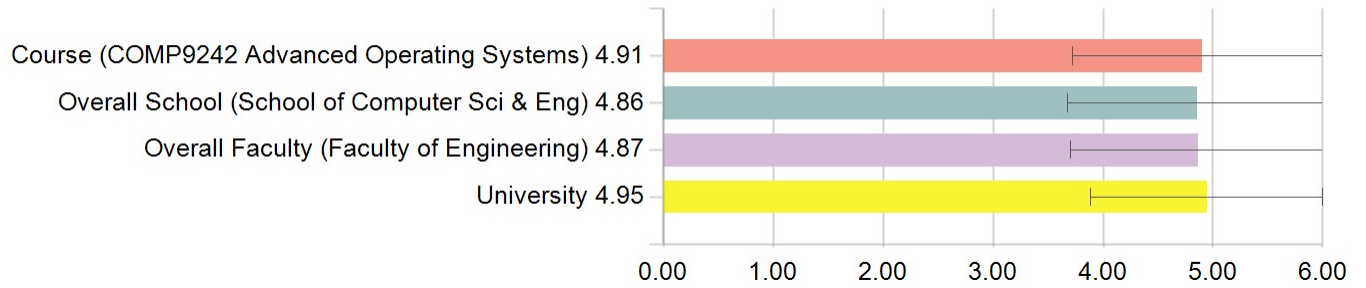
StandardDev



4. The assessment tasks were appropriate



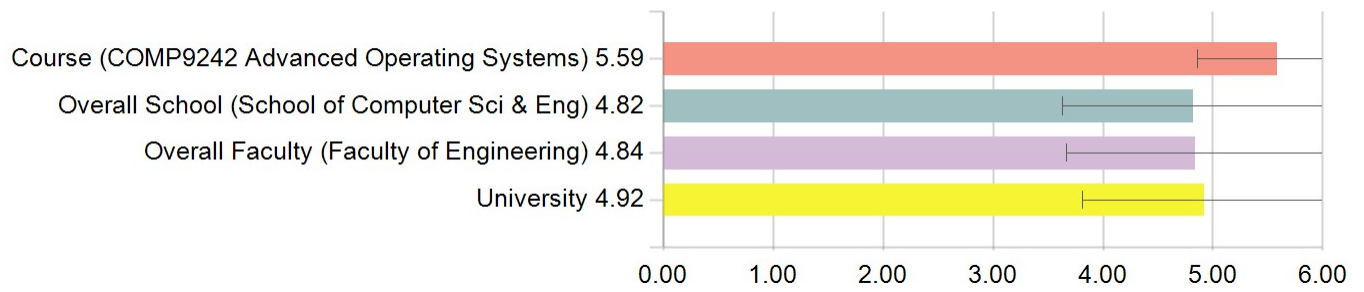
5. The amount of assessment was appropriate



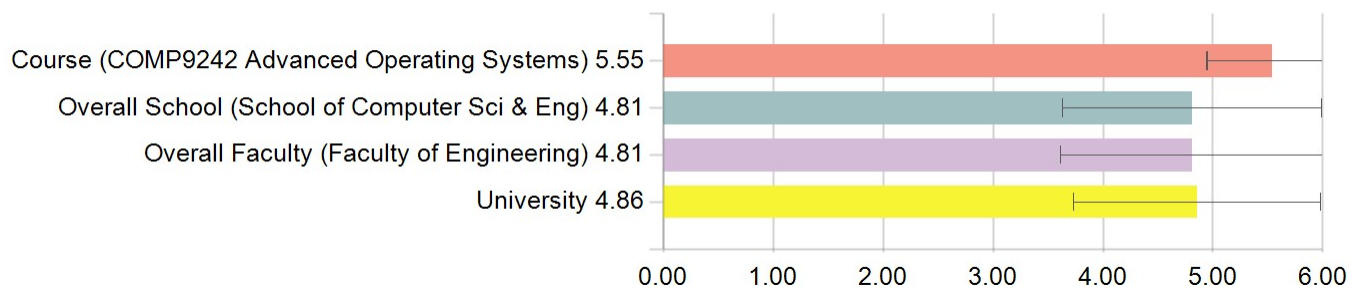
Comparison Statistics (continued)

Mean (average student responses between 1 and 6) and StandardDev (Standard deviation of student responses) are used for comparison statistics between Course, School, Faculty and University.

6. Overall I was satisfied with the quality of the teaching



7. Overall I was satisfied with the quality of the course





Raw Comment Data

What were the best things about this course?

Comments
the original, the best. accept no substitutes.
the twelve-week intensive project; who needs a life, anyway?
working on cutting-edge research tools like seL4, with the researchers who built it.
the potpourri of lecture content covering all manner of curious interesting things.
great continuity from the first OS course, weaving ideas from there towards their logical conclusions.
Loved the project, tutors always helpful and consults everyday meant help was always available. Piazza was good as a forum and any issues were fixed and handled.
The projects. It really helped me to gain a better understanding of how operating system works.
The lectures were amazing. It was an excellent course simply for that.
The best thing about this course is obviously the project. I learned a lot by being able to implement the various parts of an operating system.
I liked having the weekly milestones to ensure that we stayed on track and didn't far too fall behind.
Everything, tbh. The project is very fun. There's lots of support from staff / tutors / etc. The course has been challenging. Some of the lectures have been extremely interesting (the rest were still interesting). Beer o'clock. — Since there's no option to give feedback on the "guest" lecturers, I'll write it here instead: I found Anna's lecture very interesting and I learned a lot from it, she makes a very good lecturer. I also really enjoyed Peter's lecture on Linux internals, especially the "history lesson" at the start (I already knew a bit about things like the Lions book, but it was really cool seeing how everything fit together). (I missed Ihor's lecture because I was sick, so I can't comment on that...)
The lecture content was interesting, engaging, and covered a great array of topics. That each topic was presented by lecturers with much relevant experience in industry and research made the topics far more engaging and allowed for the presentation to be far more well informed.
The project tested our knowledge and skills to a great depth, the likes of which is achieved by few other courses. This experience allowed us to really test the extend of our understanding of course content.
project is a great experience, lectures are very impressive.
The project: it was super satisfying making a thing over the whole semester, and watching it get more and more complete as time went on.
I love the lectures as the lecturers are all at the forefront of their respective fields. They have interesting content.
The assignment is great as well and a good opportunity to learn a lot. It is tough but that is why you have an opportunity to learn a lot.
Project was challenging
Its been probably said many times, but the project was just awesome. I learnt so much from it!
Teaches you a ridiculous amount of things, very challenging but rewarding
How much freedom we had over how we design our code, the structure of the project, how much feedback we got along the way.
Getting to build your own OS, and working with a large codebase.
<ul style="list-style-type: none"> – Getting to use seL4. – Every lecture was insightful, and possible for me to understand. This was probably helped by the fact that I had actually attended lectures for the prerequisite course. – Course's transition to new hardware and mainline kernel seemed smooth enough. The two notable bugs were fixed very early on. – The tutors I interacted with during consults were quite helpful.
The content was very interesting as well as the project.

What could be improved?

Comments

let us have a twelve or thirteen week version of this course!

Add more material about linking and calling convention of the compilers. Add some material about the process of system bootstrapping phase and how boot-loader works.

The assignment wasn't related at all to the lecture content. It seemed like I was doing two different courses.

In some ways the course seems like it was designed to simply be difficult instead of being designed for students to learn. There's definitely a disconnect between course coordinators and students.

– Extended OS was not enough preparation.

– Questions on Piazza were ignored on weekends with weekly milestones due on Tuesday. If you need help you won't receive it until Monday by which point you might not have enough time to fix your submission.

Maybe milestone submissions could be on a Friday instead?

– seL4 documentation is poor and in many places lacking. Tutors were helpful but they didn't always have the answers (understandable).

It might be good to spend 20–30 minutes each week during the lecture simply talking about what's required for the current milestone and what features of seL4 can be used to complete it.

– Weekly milestones that build on each other don't foster a good learning environment.

* If you get stuck on one milestone you'll lose the opportunity to learn about the content in later milestones.

* The workload combined with weekly milestones ensures that it can't be completed individually (under a full time load) in the event that you get a bad / busy partner.

The course should either be weighted differently, have assessments weighted differently, have code provided after each milestone, or provide alternative assignments for individual submissions.

I personally felt that there is much that could be improved in this course.

1. My biggest gripe with this course is that the support from the teaching staff is very lacking. I felt like the directions for the project were very lacking and we ended up wasting lots of time because we either misunderstood things in the directions or because they weren't explained very well. This course is challenging enough and has too much work as it is. Just one as one example, I honestly felt like Milestone 0 was one of the more challenging milestones, which looking back is ridiculous because the later milestones are obviously much more complicated. It would be nice if even a portion of the first lecture could be an introduction/walkthrough for the codebase and we could, as a class, talk about the more important and relevant parts. Having us just familiarize ourselves with the codebase by telling us to read the manual seems lazy and ineffective, especially considering the manual is not very well written.

2. There is too much of a disconnect between the lecture material and the project. It almost seems like two different, unrelated courses.

3. Three hour lecture block on a Friday is pretty terrible. Who can stay awake that long? I'm sure most students would prefer lecture being split up over the week. Not very good for student learning.

4. Lecture recordings had poor audio quality. This is fairly important because as stated above, students tend to miss a lot of information over a 3 hour lecture. Being able to go back and review is an important way to learn for me personally, and the poor audio quality can be frustrating.

5. My partner and I fell behind a week on one of the earlier milestones (despite spending many late nights working on AOS) and we never caught up. Even though we were able to keep the pace and finish the subsequent milestones on a weekly basis, we were still docked a point every week because we fell behind once, and couldn't make it up by completing two weeks worth of work in one week. This was a bit frustrating and stressful.

Overall, I think that the main takeaway should be that the instructors should do a better job the first couple of weeks to ensure that students succeed in the course. I was seriously considering dropping the course in the first couple of weeks because I felt that the amount of work was overwhelming and staff support was lacking. But looking back, the first couple milestones were very simple and small in scope and there was just a lot of confusion and misunderstanding. I'm definitely glad I didn't drop the course and made it all the way to the end, but the instructors could certainly do a lot better in making the first couple of weeks more bearable.

Comments

I came into this class because I enjoyed the intro OS class so much, from start to finish. Unfortunately, I cannot say the same for this class.

Sometimes the lectures felt a bit long, and it was hard to stay focused for the full three hours (even when the content was interesting)

I have no idea how you'll make the course work in 10 weeks rather than 13, good luck...

The project is an immense workload and with the changes coming with 3+, the consistency and level of the workload makes it hard to work around other courses, assignments, and projects. I think the workload from the project should be reduced (however doing so would lead to some key topics not being covered in as much depth as they perhaps should).

lack of support at projects since the system is changed from last years and the answers at piazza is not very helpful.

Some of the libraries we were given were pretty terrible. Even in week 12, the TCP connection (and the network stack in general) was often flaky. Sometimes we'd have weird bugs, like NFS just... not writing to the pagefile. It probably would've been helpful to document known issues more clearly (for instance, it seemed to me like picotcp is very prone to races – a note of that would've been useful)

Also three hour lectures on Friday afternoons suck. Even moving them earlier on Friday would've been (slightly) better, but I guess with trimesters it's going to change to 2x2 hour lectures.

My only complaint is that some information on the website is out dated and should be updated.

For example in the last milestone with elf loading, it states that the elf file can be read from the first 4096 bytes. But in fact this year you had to read in more than that to grab the syscall table. I didnt know this and i found it very difficult to complete and it cost me 1 week.

Demand paging sucks. Remove demand paging.

Networking, libserial, libnfs, oh so many bugs.

Maybe make the filesystem local (emmc or sd card).

Demand paging might be less bad if it was local. There's a reason swap/page files generally aren't supported over a network.

Making it worth more units of credit (such as 12, or 2 courses worth). Currently, it is just too large, and almost expects you to either be doing other ridiculously easy courses, or to be underloading. If you're not doing either of these things, you cannot give the course a proper attempt as there is just simply too much time that is required from it.

There are a lot of "gotchas" that you find out about kindof later into the semester, which should have been covered earlier. Also, some of the milestones seem underspecified, making it harder to reason about what kind of tests will be run, for instance the process model and how long a process status should stick around for. Also, having to implement paging through libnfs is a bit insane.

A little more guidance with the milestones, completely working skeleton code before the semester begins.

Preface: I had some disillusion once I realised that the project was to build a monolithic server on top of seL4, but in hindsight, I don't know whether a multi-server AOS project can meaningfully be done within a single semester (or trimester) by most students. If placing Ethernet/UART drivers as well as students writing the clock driver as separate processes can be done without giving away too much of the process management milestone, it might be a reasonable idea.

– I found it difficult to navigate the manual in order to find the information I need. I'm not sure how to improve this, but it might be a good idea to make a concise version of the manual with just a list of library functions.

– I found myself struggling to navigate header files. This is probably my fault for relying on grep and not ctags...

– I also found myself trying to navigate and understand the scope of existing SOS code and not doing a very good job. A brief document explaining what is already happening might help students make progress more efficiently.

– Generally: would prefer more documentation which is easier to navigate and understand.

– Very minor gripe: plenty of spelling mistakes in comments in provided code.

Regarding non-project issues:

– The exam is obviously a critical analysis exam. I think this is great, but it could be helpful to have a small amount of explicit preparation work to do much earlier than Week 13, other than simply reading papers, since this is less about the exam and more about imparting the ability to do analysis of OS design and science more effectively.

The 3 hour lecture slot is a bit too long and personally I found it hard to pay attention and stay awake.